

3100096007622

PERPUSTAKAAN ITS	
Tgl. Terima	22 07 1994
Terima Dari	H
No. Agenda Fcp.	2658

**PERENCANAAN DAN PEMBUATAN  
PENGHITUNG KESALAHAN  
KWH METER  
DENGAN MENGGUNAKAN  
MIKROKONTROLER MC 68705**

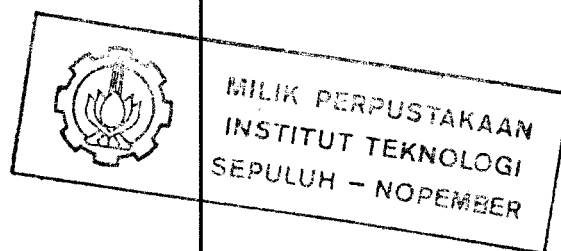
RSE  
621.391.6  
Vtd  
p-1  
1994



OLEH:

**KUSNO UTOMO**

**NRP. 291 220 1806**



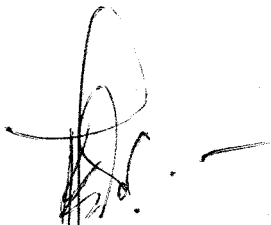
**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1994**

**PERENCANAAN DAN PEMBUATAN  
PENGHITUNG KESALAHAN  
KWH METER  
DENGAN MENGGUNAKAN  
MIKROKONTROLER MC 68705**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar  
Sarjana Teknik Elektro  
Pada  
Bidang Studi Teknik Sistem Tenaga  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui / Menyetujui  
Dosen Pembimbing**



**( Ir. TEGUH YUWONO )**

**NIP. 130 604 244**



**( Dr. Ir. SUSILO MATAIR )**

**NIP. 130 676 227**

**SURABAYA  
AGUSTUS, 1994**

## ABSTRAK

Perkembangan ilmu pengetahuan dan teknologi telah banyak menawarkan kemudahan-kemudahan bagi manusia dalam menjalankan aktifitasnya, baik berupa pekerjaan ringan maupun pekerjaan yang kompleks.

Pemakaian mikrokontroler juga telah berkembang dalam berbagai bidang, terutama dalam dunia industri. Banyak sekali kegunaan mikrokontroler ini, sehingga perkembangannya sangat pesat.

Dalam penyaluran energi listrik untuk setiap pemakaian perlu dicatat atau direkam, sehingga akan diketahui berapa pemakaian energi listrik dalam jangka waktu tertentu. Mengingat akan pentingnya hal ini, maka diperlukan kWh meter yang ketelitiannya memenuhi standar yang telah ditentukan, karena akan menyangkut berapa besarnya biaya yang harus dibayar oleh konsumen.

Pengecekan ketelitian kWh meter agar sesuai dengan standar yang telah ditentukan oleh PLN, biasanya dilakukan secara manual. Untuk memudahkan pembacaan kesalahan kWh meter tersebut, maka akan lebih mudah apabila kesalahannya dapat terbaca secara langsung (*displayed*).

Dalam pembuatan alat penghitung kesalahan kWh meter agar kesalahannya dapat terbaca secara langsung adalah dengan menggunakan IC mikrokontroler MC 68705 P3 produksi Motorola, dan didukung dengan kWh meter standar merek Enertec dengan tipe E4Y21E.

# **KATA PENGANTAR**

Puji syukur penyusun panjatkan kehadiran Allah S.W.T. atas segala rahmat dan hidayah-Nya yang telah dilimpahkan, sehingga penyusun dapat menyelesaikan penyusunan Tugas Akhir yang berjudul :

## **PERENCANAAN DAN PEMBUATAN PENGHITUNG KESALAHAN KWH METER DENGAN MENGGUNAKAN MIKROKONTROLER MC 68705**

Tugas Akhir dengan beban 6 S.K.S. (Satuan Kredit Semester) ini merupakan salah satu persyaratan untuk memperoleh gelar sarjana teknik elektro pada bidang studi Teknik Sistem Tenaga, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dengan menyadari akan keterbatasan kesempatan, serta kemampuan yang ada dalam penyusunan Tugas Akhir ini, penyusun berharap semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

Surabaya, Agustus 1994

Penyusun

## UCAPAN TERIMA KASIH

Dengan telah selesainya penyusunan Tugas Akhir ini, maka penyusun ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Bapak DR. Ir. Susilo Matair dan Bapak Ir. Teguh Yuwono, selaku dosen pembimbing yang telah meluangkan waktu disela-sela kesibukannya untuk memberikan bimbingan serta saran-saran yang berguna, hingga penyusun dapat menyelesaikan penyusunan Tugas Akhir.
2. Bapak DR. Ir. Ontoseno Penangsang, selaku dosen wali yang telah banyak memberikan dorongan semangat, hingga penyusun dapat menyelesaikan penyusunan Tugas Akhir ini.
3. Bapak DR. Ir. Moch. Salehudin M.Eng.Sc., selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya, yang telah banyak memberikan bantuan, hingga penyusunan Tugas Akhir ini dapat terlaksana .
4. Bapak Ir. Sidarjanto, selaku koordinator bidang studi Teknik Sistem Tenaga yang telah membantu, hingga penyusunan Tugas Akhir ini dapat terlaksana hingga selesai.
5. Seluruh dosen Teknik Elektro, yang telah berkenan memberikan ilmu yang sangat bermanfaat bagi penyusun untuk digunakan sebagai sarana penunjang dalam menyelesaikan penyusunan Tugas Akhir.

6. Kepala Siralab PLN Distribusi Jawa Timur, beserta staf yang telah membantu memberikan fasilitas untuk melakukan percobaan dan pengetesan alat, demi terselesaikannya penyusunan Tugas Akhir ini.
7. Ayah, Ibu, Istri, Anak, serta saudara-saudara yang telah memberikan dorongan semangat yang amat dibutuhkan penyusun dalam rangka penyelesaian Tugas Akhir ini.
8. Semua pihak yang telah turut membantu dalam penyelesaian Tugas Akhir ini.

Semoga Allah S.W.T. memberikan berkah rahmat dan pahala yang setimpal atas jasa-jasanya yang telah diberikan.

# DAFTAR ISI

BAB	HAL.
JUDUL .....	i
PENGESAHAN .....	ii
ABSTRAK .....	iii
KATA PENGANTAR .....	iv
UCAPAN TERIMA KASIH .....	v
DAFTAR ISI .....	vii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xiii
I. PENDAHULUAN .....	1
I.1. Latar Belakang .....	1
I.2. Pokok Pembahasan .....	2
I.3. Sistematika Pembahasan .....	2
I.4. Tujuan Pembahasan .....	3
I.5. Relevansi .....	3
II. PENGUKURAN DAYA DAN WATT HOUR METER .....	4
II.1. Pengukuran Daya dan Faktor Kerja .....	4

II.1.1. Pengukuran Daya Satu Phasa .....	8
II.1.2. Pengukuran Daya Tiga Phasa .....	8
II.1.2.1. Pengukuran Daya Tiga Phasa Dengan Dua Watt Meter .....	8
II.1.2.2. Pengukuran Daya Tiga Phasa Dengan Tiga Watt Meter .....	10
II.2. Pengukuran Daya Reaktif .....	11
II.3. Pengukuran Energi .....	12
II.4. Watt Hour Meter .....	13
II.4.1. Watt Hour Meter Induksi .....	13
II.4.1.1. Penyesuaian Phasa .....	16
II.4.1.2. Penyesuaian Beban Berat .....	17
II.4.1.3. Penyesuaian Beban Ringan .....	18
II.4.1.4. Mengelakkan Putaran Pada Beban Kosong .....	19
II.4.2. Karakteristik kWh Meter Induksi .....	20
II.4.3. Watt Hour Meter <i>Solid State</i> .....	21
III. MIKROKONTROLER .....	23
III.1. Uraian Umum .....	23
III.2. Chip Tunggal MC 68705 P3 .....	24
III.2.1. Sifat dan Karakteristik Chip Tunggal MC 68705 P3 .....	24



III.2.2. Memori Map MC 68705 P3 .....	29
III.2.3. Perangkat Lunak Chip Tunggal MC 68705 P3 .....	30
III.2.4. Perangkat Keras Chip Tunggal MC 68705 P3 .....	32
III.2.5. Teknik Pemrograman .....	44
III.3. Program Simulasi 6805 AVSIM .....	46
III.3.1. Pelaksanaan Simulasi dan Perintah pada Simulator .....	47
IV. PERENCANAAN DAN PEMBUATAN .....	49
IV.1. Blok Perencanaan .....	51
IV.2. Perangkat Keras .....	54
IV.2.1. Sensor .....	55
IV.2.2. Mode Switch .....	56
IV.2.3. Konstanta DIP Switch .....	57
IV.2.4. Mikrokontroler .....	59
IV.2.5. Tampilan .....	62
IV.3. Perangkat Lunak .....	67
IV.3.1. Diagram Alir .....	67
IV.4. Cara kerja penghitung kesalahan kWh meter .....	71
V. KESIMPULAN .....	75
DAFTAR PUSTAKA .....	76
LAMPIRAN .....	78

# DAFTAR GAMBAR

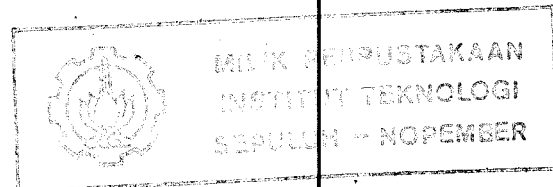
GAMBAR	HAL.
2.1. BEBAN TAHANAN VS ARUS BOLAK-BALIK .....	5
2.2. BEBAN INDUKTIF VS DAYA ARUS BOLAK-BALIK ...	7
2.3. BEBAN UMUM VS DAYA ARUS BOLAK-BALIK .....	7
2.4. CARA PENGUKURAN DENGAN WATT METER .....	8
2.5. PENGUKURAN DAYA TIGA PHASA DENGAN METODA TIGA KAWAT .....	9
2.6. PENUNJUKKAN WATT METER DENGAN METODA DUA WATT METER .....	10
2.7. PENGUKURAN DAYA TIGA PHASA DENGAN METODA EMPAT KAWAT .....	11
2.8. PENGUKURAN DAYA REAKTIF .....	11
2.9. ELEMEN DASAR WATT HOUR METER INDUKSI .....	12
2.10. PRINSIP KWH METER INDUKSI .....	15
2.11. ARUS EDDY PADA KEPINGAN .....	15
2.12. PRINSIP PENGATUR PHASA .....	16
2.13. PRINSIP PENGATUR BEBAN BERAT .....	18
2.14. PRINSIP PENGATUR BEBAN RINGAN .....	19

2.15. KARAKTERISTIK KWH METER INDUKSI .....	21
2.16. DIAGRAM BLOK PRINSIP KERJA KWH METER <i>SOLID</i> <i>STATE</i> .....	22
3.1. BLOK DIAGRAM MC 68705 P3 .....	25
3.2. ARSITEKTUR REGISTER MC 68705 P3 .....	26
3.3. MEMORI MAP MC 68705 P3 .....	30
3.4. BLOK DIAGRAM TIMER .....	34
3.5. RANGKAIAN EKSTERNAL INTERRUPT .....	42
3.6. DIAGRAM ALIR PELAYANAN INTERRUPT .....	43
4.1. PENERAAN KWH METER INDUKSI SECARA MANUAL .	50
4.2. BLOK DIAGRAM PENGHITUNG KESALAHAN KWH ME- TER .....	51
4.3. RANGKAIAN SENSOR .....	55
4.4. BENTUK OUTPUT SENSOR .....	55
4.5. RANGKAIAN MODE SWITCH .....	56
4.6. RANGKAIAN KONSTANTA DIP SWITCH .....	58
4.7. RANGKAIAN MIKROKONTROLER .....	62
4.8. RANGKAIAN TAMPILAN MENU PENGETESAN .....	63
4.9. RANGKAIAN TAMPILAN PROSENTASE KESALAHAN .	63
4.10. RANGKAIAN TAMPILAN TINGKAT KESALAHAN ....	64

4.11. RANGKAIAN SENSOR DAN PENGKONDISI SINYAL . . .	65
4.12. RANGKAIAN UTAMA . . . . .	66
4.13. DIAGRAM ALIR PENGHITUNG KESALAHAN KWH ME- TER . . . . .	68
4.14. POSISI SETTING DIP SWITCH . . . . .	72
4.15. SAMPLING PULSA . . . . .	73

# DAFTAR TABEL

TABEL	HAL.
2.1. PROSENTASE BATAS KESALAHAN KWH METER INDUKSI	20
3.1. HUBUNGAN TIN & TIE .....	38
3.2. <i>PRESCALER SELECT BIT</i> .....	39
3.3. FUNGSI PIN I/O .....	41
3.4. PRIORITAS INTERRUPT .....	42
3.5. <i>PROGRAMMING CONTROL REGISTER</i> .....	45
4.1. KONFIGURASI MENU .....	73



# BAB I

## PENDAHULUAN

### I.1. Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi telah banyak menawarkan kemudahan-kemudahan bagi manusia dalam menjalankan aktifitasnya, baik berupa pekerjaan ringan maupun pekerjaan yang kompleks.

Pemakaian mikroprosesor juga telah berkembang pesat dalam berbagai bidang, terutama dalam dunia industri. Banyak sekali kegunaan mikroprosesor ini, sehingga perkembangannya sangat pesat.

Dalam penyaluran energi listrik untuk setiap pemakaian perlu dicatat atau direkam, sehingga akan diketahui berapa pemakaian energi listrik dalam jangka waktu tertentu. Mengingat akan pentingnya hal ini, maka diperlukan kWh meter yang ketelitiannya memenuhi standar yang telah ditentukan, karena akan menyangkut berapa besarnya biaya yang harus dibayar oleh konsumen.

Pengecekan ketelitian kWh meter agar sesuai dengan standar yang telah ditentukan oleh PLN, biasanya dilakukan secara manual. Untuk memudahkan pembacaan kesalahan kWh meter tersebut, maka akan lebih mudah apabila kesalahannya dapat terbaca secara langsung (*displayed*).

Dalam tugas akhir ini akan dibahas mengenai Perencanaan dan pembuatan alat penghitung kesalahan kWh meter berbasis IC mikrokontroler MC 68705 P3, dan didukung dengan kWh meter standar. Dengan alat ini

diharapkan dapat mengetahui kesalahan kWh meter, baik untuk peneraan maupun pengontrolan di lapangan, sesuai dengan yang diharapkan.

## **I.2. Pokok Pembahasan**

Sesuai dengan judul tugas akhir ini, yaitu "PERENCANAAN DAN PEMBUATAN PENGHITUNG KESALAHAN KWH METER DENGAN MENGGUNAKAN MIKROKONTROLER MC 68705", maka pokok pembahasan di dalam Tugas Akhir ini hanya dibatasi pada pembuatan alat penghitung kesalahan kWh meter, yang ditujukan untuk membantu menghitung kesalahan kWh meter yang sedang di tera atau di uji, dengan menggunakan mikrokontroler MC 68705 P3 sebagai komponen utama, beserta bahasa program *Assembly* produksi *AVOCET System, Inc.*

## **I.3. Sistematika Pembahasan**

Secara garis besar, pembahasan di dalam Tugas Akhir ini dapat dibagi menjadi tiga bagian besar, yaitu :

- a. teori penunjang.
- b. perencanaan, pembuatan.
- c. kesimpulan.

Dasar-dasar teori penunjang yang dituliskan dalam Tugas Akhir terbagi menjadi dua bab, yaitu pada bab II dibahas mengenai pengukuran daya, dan cara kerja kWh meter, serta karakteristiknya , sedangkan pada bab III dibahas tentang mikrokontroler, dan cara pemrogramannya.

Pada bab IV akan dibahas mengenai perencanaan, pembuatan alat yang akan digunakan untuk menghitung kesalahan kWh meter.

Akhir dari keseluruhan pembahasan dalam Tugas Akhir ini terdapat pada bab V atau bab kesimpulan, yang berisi beberapa kesimpulan yang dapat diambil dari pembahasan-pembahasan yang telah dilakukan di dalam bab-bab sebelumnya yang terdapat dalam penyusunan Tugas Akhir ini.

#### **I.4. Tujuan Pembahasan**

Dalam merencanakan dan membuat alat penghitung kesalahan kWh meter, dengan menggunakan IC mikrokontroler MC 68705 P3 yang berfungsi sebagai komponen utama, serta didukung dengan komponen-komponen lainnya sebagai penunjang.

#### **I.5. Relevansi**

Dengan alat penghitung kesalahan kWh meter yang dapat menampilkan kesalahannya secara langsung, diharapkan dapat membantu Perusahaan Umum Listrik Negara (PLN) dan atau pihak yang memerlukannya, dengan demikian kesalahan kWh meter induksi dapat segera diketahui baik untuk peneraan maupun pengontrolan di lapangan, karena alat ini mudah cara pengoperasiannya.



## BAB II

### PENGUKURAN DAYA DAN WATT HOUR METER

#### II.1. Pengukuran Daya dan Faktor Kerja

Pada jala-jala arus bolak-balik, daya yang dipakai dalam beban pada saat dimana tegangan beban dan arus beban adalah  $v$  dan  $i$ , maka harga sesaat dari daya dapat dinyatakan sebagai berikut :

$$p = v \cdot i$$

Bila sekarang tegangan dapat dinyatakan sebagai fungsi sinus dan ditulis sebagai :

$$v = V_m \sin \omega t$$

Bila tahanan beban adalah  $R$ , maka arus beban dapat dinyatakan sebagai :

$$i = \frac{V_m}{R} \sin \omega t = I_m \sin \omega t$$

dengan demikian daya  $p$  dapat dinyatakan sebagai :

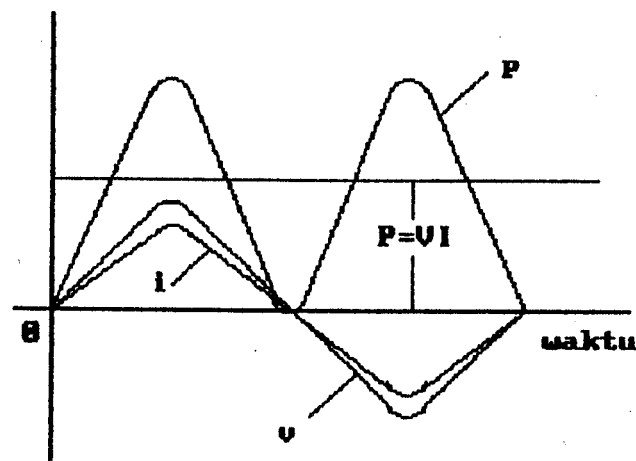
$$p = V_m I_m \sin^2 \omega t$$

dengan menggunakan persamaan :

$$\sin^2 = \frac{1}{2} (1 - \cos 2\alpha)$$

didapatkan :

$$p = \frac{V_m I_m}{2} (1 - \cos 2\omega t)$$



GAMBAR 2.1  
BEBAN TAHANAN VS ARUS BOLAK-BALIK

Hubungan tersebut diatas dapat dilukiskan dalam gambar 2.1, sesuai dengan definisi dari harga efektif, maka harga rata-rata dari daya  $p$  melalui satu perioda, yaitu harga rata-rata  $p$ , dapat dinyatakan sebagai :

$$p = \frac{V_m I_m}{2} = V I = I^2 R = \frac{V^2}{R}$$

dimana  $V$  dan  $I$  adalah harga efektif dari  $v$  dan  $i$  masing-masingnya.

Persamaan ini mempunyai bentuk yang sama dengan persamaan untuk arus searah. Maka dengan demikian  $p$  disebut sebagai daya dalam arus bolak-balik. Bila kemudian beban mempunyai elemen yang reaktif, misalkan beban dengan induktansi  $L$ , maka :

$$i = \frac{V_m}{\omega L} \sin \left( \omega t - \frac{\pi}{2} \right) = I_m \sin \left( \omega t - \frac{\pi}{2} \right)$$

dan dengan demikian akan didapatkan :

$$p = V_m I_m \sin \left( \omega t \right) \sin \left( \omega t - \frac{\pi}{2} \right)$$

$$p = \frac{V_m I_m}{2} \sin 2\omega t$$

Hubungan ini dinyatakan dalam gambar 2.2, dari gambar tersebut dapat dilihat, bila induktansi yang diberikan energi dari sumber bolak-balik, untuk setengah periode akan menyimpan energi elektromagnetik, dan mengembalikan energi tersebut pada setengah periode berikutnya. Jadi induktansi tidak mempergunakan daya secara rata-rata. Dengan kata lain, maka harga rata-rata daya dalam induktansi adalah nol.

Misalkan sekarang beban adalah kombinasi dari tahanan dan reaktansi, yaitu dinyatakan sebagai  $Z = R + jX$ , dimana  $R$  adalah tahanan dan  $X$  adalah reaktansi, maka :

$$i = I_m \sin \omega t$$

$$p = V_m I_m \sin \omega t \sin (\omega t - \phi)$$

dengan  $\tan \phi = \frac{X}{R}$ , dan akan didapat :

$$p = V I \cos \phi (1 - \cos 2\omega t) - V I \sin \phi \sin 2\omega t$$

Hubungan ini digambarkan dalam gambar 2.3, dan harga rata-rata  $P$  adalah :

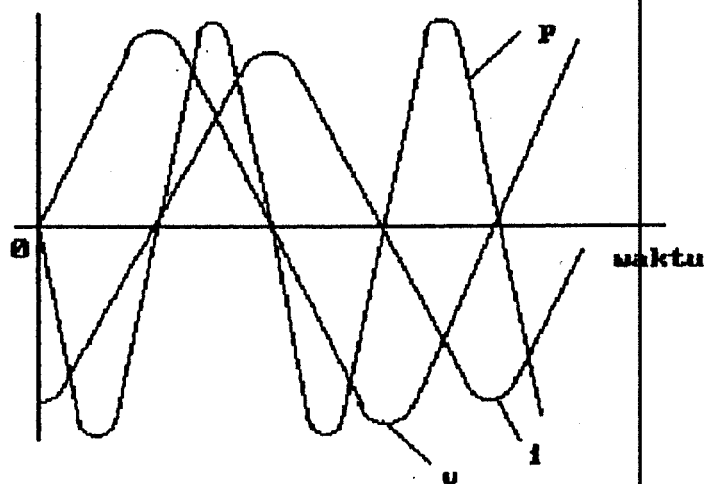
$$P = V I \cos \phi$$

Kemudian dengan mempergunakan :

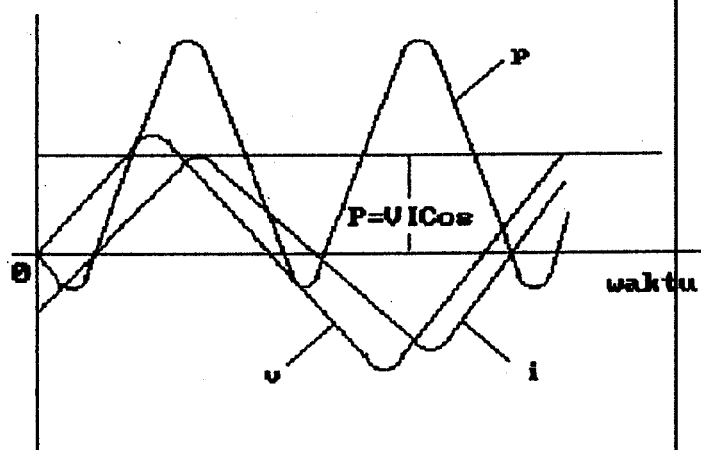
$$\cos \phi = \frac{R}{\sqrt{R^2 + X^2}}$$

akan dihasilkan :

$$P = I^2 R = \frac{V^2}{R}$$



GAMBAR 2.2  
BEBAN INDUKTIF VS DAYA ARUS BOLAK-BALIK



GAMBAR 2.3  
BEBAN UMUM VS DAYA ARUS BOLAK-BALIK

Seperti diperlihatkan diatas maka harga rata-rata dari daya dalam keadaan arus bolak-balik dengan beban yang umum, dapat dinyatakan sebagai hasil kali dari arus dan tegangan yang dinyatakan sebagai perkalian dari harga efektif dari arus dan tegangan, dengan cosinus dari perbedaan fasa antara arus dan tegangan yang melalui beban tersebut.

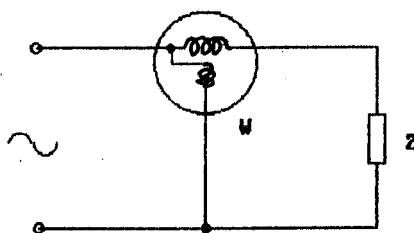
Harga dari hasil perkalian ini adalah sama dengan daya yang dipakai dalam tahanan. Jadi daya ini disebut pula daya efektif. Disamping ini, maka amplitudo harga sesaat dari daya  $V I \sin \phi$ , yang hanya merupakan daya yang ditransferkan antara sumber dengan reaktansi beban, tidak dipergunakan dan disebut daya reaktif,  $V I$  disebut daya semu, dan  $\cos \phi$ , disebut faktor kerja.

### II.1.1. Pengukuran Daya Satu Fasa

Daya satu fasa dapat diukur dengan mempergunakan satu Watt meter, gambar 2.4 memperlihatkan cara pengukuran dengan Watt meter.

Penunjukkan Watt meter  $W$  = Daya dari beban, adalah :

$$P = V I \cos \phi$$



GAMBAR 2.4  
CARA PENGUKURAN DENGAN WATT METER

### II.1.2. Pengukuran Daya Tiga Fasa

#### II.1.2.1. Pengukuran Daya Tiga Fasa Dengan Dua Watt Meter

Pengukuran daya tiga fasa dapat dilakukan dengan dua alat pengukur Watt meter satu fasa, yaitu dengan cara menjumlahkan secara aljabar

hasil penunjukkannya. Cara ini disebut metoda dengan dua alat pengukur Watt atau metoda tiga kawat. Seperti diperlihatkan pada gambar 2.5, dalam cara ini bila susunan dari phasanya adalah 1-2-3, maka bila penunjukkan dari alat-alat pengukur watt adalah  $W_1$  dan  $W_2$ , maka akan didapat persamaan :

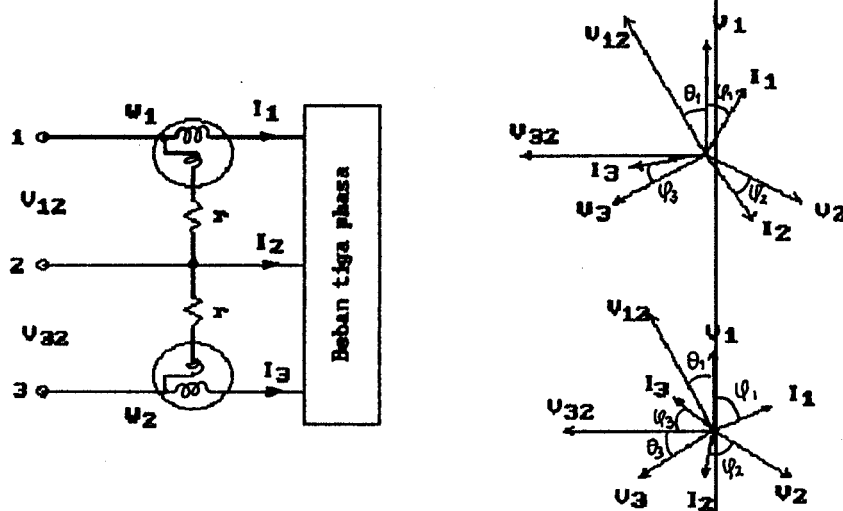
$$W_1 = V_{12} I_1 \cos (\theta_1 + \varphi_1)$$

$$W_2 = V_{32} I_3 \cos (\theta_3 - \varphi_3)$$

Bila tegangan dalam jaringan tiga phasa ini adalah seimbang, maka  $V_{12} = V_{23} = V_{31}$  dan  $\theta_1 = \theta_3 = 30^\circ$ . Sedangkan bila bebannya seimbang, maka :  $I_1 = I_2 = I$ , dan  $\varphi_1 = \varphi_3 = \varphi$ . Dengan demikian persamaan diatas menjadi :

$$W_1 = V I \cos (30^\circ + \varphi)$$

$$W_2 = V I \cos (30^\circ - \varphi)$$

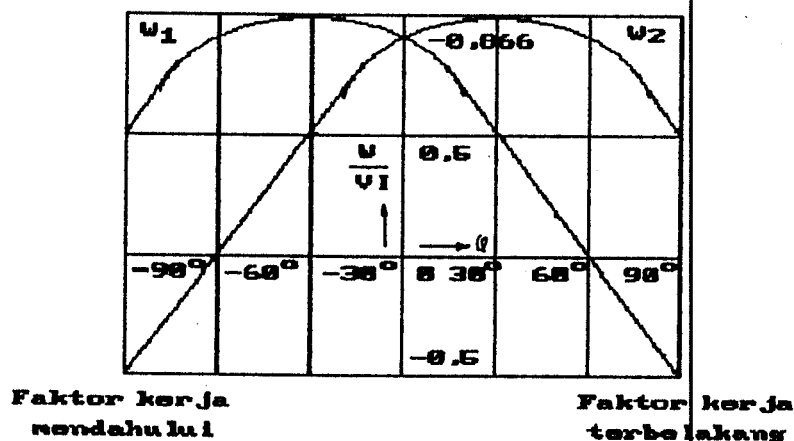


GAMBAR 2.5<sup>1)</sup>  
PENGUKURAN DAYA TIGA PHASA DENGAN METODA TIGA KAWAT

<sup>1)</sup> Soedjana Sapi'i, DR., Osamu Nishino, DR., Pengukuran dan Alat-alat Ukur Listrik, Pradnya Paramita, Cetakan ketiga, Jakarta, 1979, hal. 67.

Hubungan antara  $W_1$ , serta  $W_2$ , diperlihatkan pada gambar 2.6, dan daya totalnya adalah :

$$P_{\text{total}} = W_1 + W_2$$



GAMBAR 2.6<sup>2)</sup>  
PENUNJUKKAN WATT METER DENGAN METODA DUA WATT METER

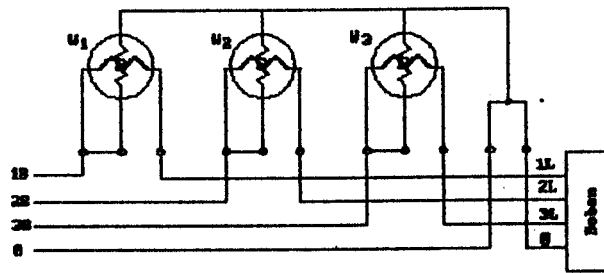
#### II.1.2.2. Pengukuran Daya Tiga Fasa Dengan Tiga Buah Watt Meter

Pengukuran daya tiga fasa juga dapat dilakukan dengan menggunakan tiga Watt meter, yaitu dengan jalan memasang masing-masing Watt meter, pada masing-masing fasanya. Cara ini disebut dengan pengukuran daya tiga fasa dengan tiga Watt meter, atau sering disebut dengan metoda empat kawat, seperti terlihat pada gambar 2.7 berikut ini.

Jadi daya tiga fasa tersebut merupakan penjumlahan dari sistem satu fasa.

$$W = (V_R I_R \cos \phi_R + V_S I_S \cos \phi_S + V_T I_T \cos \phi_T)$$

<sup>2)</sup> Ibid, hal. 67.

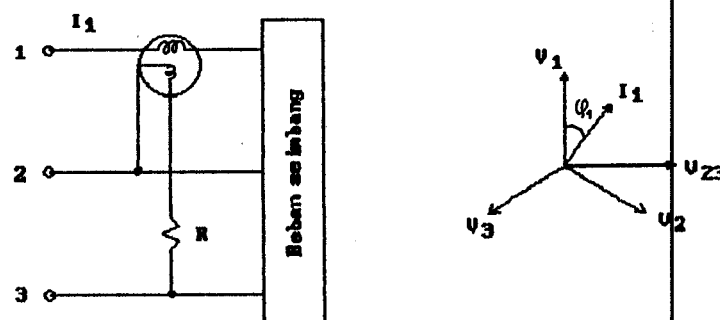


GAMBAR 2.7  
PENGUKURAN DAYA TIGA PHASA DENGAN METODA EMPAT KAWAT

## II.2. Pengukuran Daya Reaktif

Daya reaktif  $Q$  dapat dihitung dari  $V I \sin \phi$ , bila tegangan  $V$  dan arus  $I$  dan perbedaan fasa diketahui, maka daya reaktif tersebut dapat pula dihitung dari :  $Q = \sqrt{(VI)^2 - W}$ . Selanjutnya daya reaktif, dapat pula diukur dengan memberikan perbedaan fasa sebesar  $90^\circ$ , dari arus yang mengalir melalui kumparan tegangan dari alat pengukur volt terhadap tegangan jaringan.

Bila tegangan maupun beban dalam jaringan tiga fasa dalam keadaan seimbang, maka tegangan  $V_{2,3}$ , dapat ditempatkan dalam rangkaian tegangan, seperti gambar 2.8.



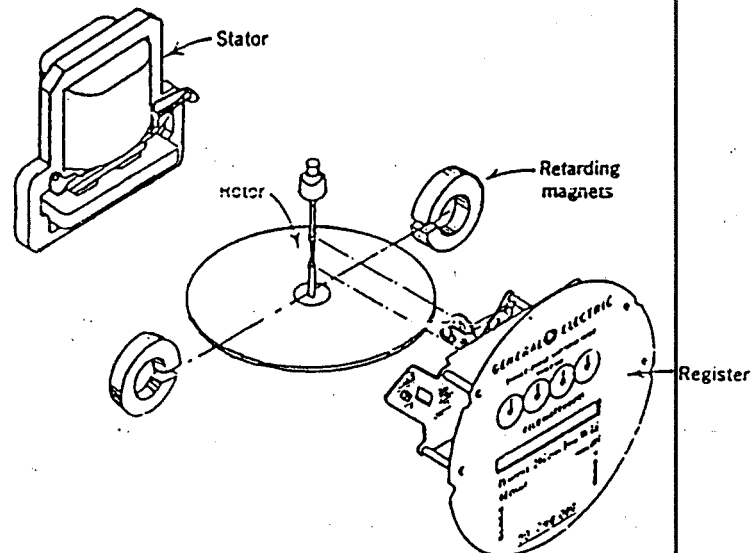
GAMBAR 2.8  
PENGUKURAN DAYA REAKTIF DARI SUATU BEBAN TIGA PHASA SEIMBANG



### II.3. Pengukuran Energi

KWh meter adalah alat pengukur energi listrik yang mengukur langsung hasil kali tegangan, arus faktor kerja, kali waktu tertentu ( $VI \cos \phi t$ ), yang bekerja padanya selama jangka waktu tertentu tersebut, hal ini berdasarkan bekerjanya induksi magnetis oleh medan magnet yang dibangkitkan oleh arus yang melalui kumparan arus terhadap kepingan (*disk*) kWh meter, dimana induksi magnet ini berpotongan dengan induksi magnetis yang dibangkitkan oleh kumparan tegangan terhadap kepingan yang sama.

Kopel putar dapat dibangkitkan terhadap kepingan karena induksi magnetis kedua medan magnet tersebut diatas bergeser fasa  $90^\circ$  satu terhadap lainnya. Hal ini dimungkinkan dengan konstruksi kumparan tegangan dibuat dalam jumlah besar gulungannya sehingga dapat dianggap induktansi murni.



GAMBAR 2.9<sup>3)</sup>  
ELEMEN DASAR WATT HOUR METER INDUKSI

<sup>3)</sup> Isaac F. Kinnard, DSc., Applied Electrical Measurement, John Weley & Son, Inc., New York, Chapman & Hall, Ltd., London, 1956, hal. 186.



## II.4. Watt Hour Meter

Alat pengukur energi listrik yang sering dikenal adalah dengan tipe induksi yang sering disebut dengan kWh meter induksi. Akan tetapi dengan berkembangnya ilmu pengetahuan terutama dalam bidang elektronika dewasa ini, maka sekarang banyak juga alat pengukur energi listrik dengan menggunakan komponen elektronika, atau sering disebut dengan kWh meter *solid state*.

### II.4.1. Watt Hour Meter Induksi

Alat ukur energi yang paling banyak digunakan pada arus bolak-balik adalah alat ukur energi tipe induksi. Alat ukur dari tipe ini mempunyai peralatan gerak yang prinsip kerjanya adalah sama dengan alat ukur dari tipe induksi seperti yang ditunjukkan dalam gambar 2.10.

Pada gambar tersebut, Cp adalah inti besi kumparan tegangan, Wp kumparan adalah tegangan, Cc adalah inti kumparan arus dan Wc adalah kumparan arus.

Arus beban I mengalir melalui Wc dan menyebabkan terjadinya fluksi  $\Phi_1$ . Wp mempunyai sejumlah lilitan yang besar dan cukup besar untuk dianggap sebagai reaktansi murni, sehingga arus Ip yang mengalir melalui Wp akan tertinggal phasanya terhadap tegangan beban dengan sudut sebesar  $90^\circ$ , dan menyebabkan terjadinya fluksi magnetik sebesar  $\Phi_2$ .

Dari gambar 2.10, terlihat bahwa, kepingan D mendapat momen gerak  $T_D$  yang berbanding lurus terhadap daya beban. Misalkan bahwa oleh pengaruh momen gerak ini, kepingan D aluminium akan berputar dengan kecepatan

n putaran. Sambil berputar ini, kepingan D akan memotong garis-garis fluksi  $\Phi_m$  dari magnet yang permanen dan akan menyebabkan terjadinya arus-arus putar yang berbanding lurus terhadap  $n\Phi_m$  didalam kepingan alumunium tersebut. Arus-arus putar ini akan memotong pula garis-garis fluksi  $\Phi_m$ , sehingga kepingan D akan mengalami momen redaman  $T_d$  yang berbanding lurus terhadap  $n\Phi_m^2$ , seperti terlihat pada persamaan dibawah ini.

$$\omega \Phi_1 \Phi_2 \sin \alpha = k V I \cos \varphi$$

Bila momen-momen  $T_d$  dan  $T_D$  ada dalam keadaan seimbang maka akan berlaku :

$$k_d V I \cos \varphi = k_m n \Phi_m^2$$

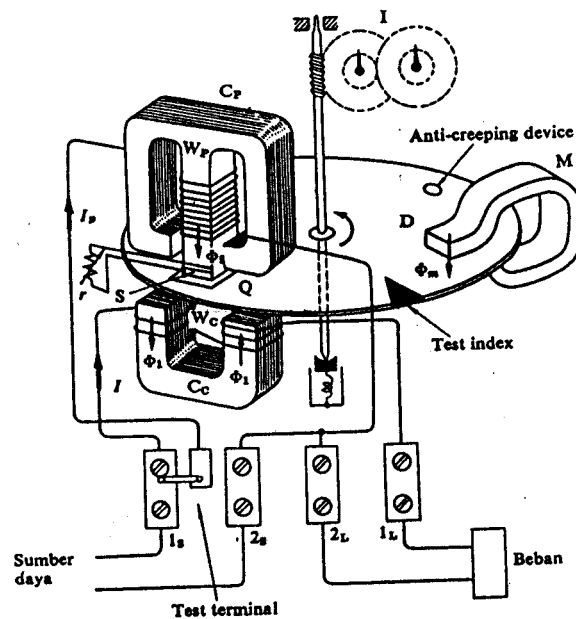
atau :

$$n = \frac{k_d}{k_m \Phi_m^2} V I \cos \varphi$$

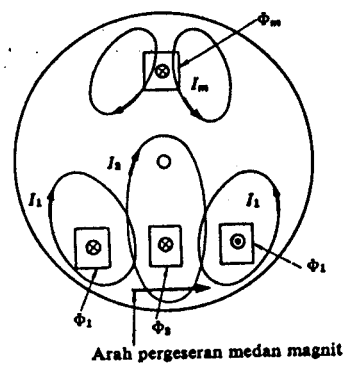
dengan  $k_d$  dan  $k_m$  sebagai konstanta. Jadi dari persamaan tersebut terlihat bahwa kecepatan putar  $n$ , dari kepingan D, adalah berbanding lurus dengan beban  $V I \cos \varphi$ , sehingga dengan demikian jumlah perputaran dari kepingan tersebut untuk suatu jangka waktu tertentu berbanding dengan energi yang akan diukur untuk jangka waktu tertentu tersebut.

Untuk memungkinkan pengukuran, maka jumlah perputaran dari kepingan D ditransformasikan melalui sistem mekanis tertentu, kepada alat penunjuk roda-roda angka. Transformasi dari kecepatan putar biasanya diadakan, sehingga roda-roda angka tersebut berputar lebih lambat dibandingkan dengan kepingan D.

Dengan demikian maka alat penunjuk atau roda-roda angka akan menunjukkan energi yang diukur dalam kWh, setelah melalui kalibrasi tertentu.



GAMBAR 2.10<sup>4)</sup>  
PRINSIP KWH METER INDUKSI



GAMBAR 2.11<sup>5)</sup>  
ARUS EDDY PADA KEPINGAN

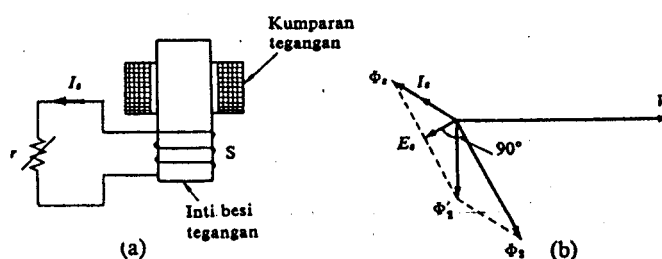
<sup>4)</sup> Ibid, hal. 72.

<sup>5)</sup> Ibid, hal. 72.

#### II.4.1.1. Penyesuaian Fasa

Agar pada kepingan dapat diberikan suatu momen yang berbanding lurus terhadap daya beban, maka  $\Phi_2$  harus tertinggal  $90^\circ$  terhadap  $V$ . Akan tetapi dalam prakteknya sudut fasa ini adalah lebih kecil dari  $90^\circ$ , yang disebabkan oleh adanya tahanan-tahanan dan kerugian-kerugian besi pada inti dari kumparan tegangan  $W_p$ . Untuk mengkompensasi hal ini, diperlukan suatu penyesuaian fasa yang ditempatkan pada kumparan tersebut.

Hal ini dicapai dengan melilitkan kumparan  $F$ , dengan beberapa lilitan melalui kumparan tegangan dan menghubungkannya dengan suatu tahanan  $R$ , seperti diperlihatkan pada gambar 2.13. Dari gambar tersebut terlihat, bahwa arus  $I_s$  yang mengalir disebabkan oleh fluksi magnetis  $\Phi_2$ , sebelum penyesuaian dibuat yang membangkitkan fluksi magnetis  $\Phi_s$ , kemudian menyebabkan fluksi kombinasi  $\Phi'_2$  dari  $\Phi_2$  dan  $\Phi_s$ , agar mempunyai fasa yang tertinggal terhadap  $V$ , dengan sudut  $90^\circ$ .



GAMBAR 2.12<sup>6)</sup>  
PRINSIP PENGATUR PHASA

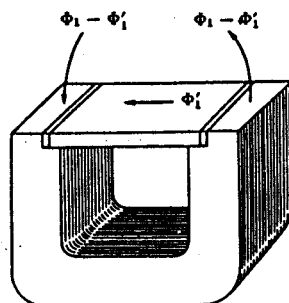
<sup>6)</sup> Ibid, hal. 73.

#### II.4.1.2. Penyesuaian Pada Beban Berat

Kepingan D pada saat berputar akan memotong fluksi-fluksi  $\Phi_1$  dan  $\Phi_2$  selain dari  $\Phi_m$ , akan membangkitkan momen-momen  $k_1 n \Phi_1^2$  dan  $k_2 n \Phi_2^2$ . Momen-momen tersebut akan bekerja berlawanan arahnya dari perputaran, yang menyebabkan perlambatan, sehingga dengan demikian suatu kesalahan negatif akan bertambah dengan bertambah besarnya  $\Phi_1$  dan atau  $\Phi_2$ . Dalam sistem tenaga, dimana alat pengukur energi ini dipergunakan, tegangan pada beban adalah hampir-hampir tetap, dengan demikian menyebabkan pula  $\Phi_2$  hampir-hampir tetap. Akan tetapi arus-arus beban akan bervariasi sangat lebar, yang pula menyebabkan terjadinya variasi dari  $\Phi_1$ . Jadi dengan demikian, pada beban berat kesalahan negatif yang disebabkan oleh  $k_1 n \Phi_1^2$  akan terjadi. Untuk mengurangi kesalahan ini  $\Phi_1$  dibuat kecil,  $\Phi_2$  besar dan perputaran  $n$  kecil. Disamping ini suatu shunt magnetis ditempatkan dalam inti kumparan arus, seperti diperlihatkan dalam gambar 2.13.

Pada saat arus-arus beban  $I$  kecil dan demikian pula  $\Phi_1$  kecil, maka shunt magnetis ini akan memungkinkan fluksi magnetis  $\Phi'_1$  yang merupakan suatu fraksi dari  $\Phi_1$ , yang mengalir melaluinya. Jadi fluksi magnetis yang disebabkan oleh arus dan memotong kepingan D, akan berkurang dari  $\Phi_1$  menjadi  $(\Phi_1 - \Phi'_1)$  akan tetapi pada saat  $I$  besar maka  $\Phi'_1$  akan bertambah besar pula sampai dengan  $\Phi'_{1m}$  kejenuhan dari fluksi magnetis terjadi, dan dengan demikian tidak akan memungkinkan mengalirnya fluksi-fluksi tersebut melalui shunt itu. Dalam hal ini pada beban berat, fluksi arus yang memotong

kepingan D dan dengan demikian pula momen gerak yang akan dihasilkannya akan bertambah secara perbandingan lebih besar terhadap arus, sehingga suatu kompensasi untuk kesalahan negatif pada beban berat akan terjadi.



GAMBAR 2.13<sup>7)</sup>  
PRINSIP PENGATUR BEBAN BERAT

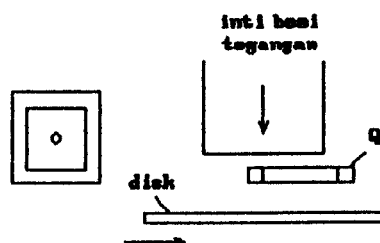
#### II.4.1.3. Penyesuaian Beban Ringan

Bila kepingan D berputar, maka momen-momen gerak mekanis akan terjadi dan menyebabkan pula kesalahan-kesalahan negatif. Kesalahan ini akan lebih penting pada beban-beban ringan, yaitu bila arus beban kecil. Untuk mengkompensasi kesalahan ini, penyesuai pada beban ringan ditempatkan seperti pada gambar 2.14. Q pada gambar tersebut adalah suatu cincin tembaga yang pendek, yang ditempatkan antara kumparan tegangan dan kepingan dalam posisi yang agak miring pada arah perputaran.

Dengan pengaturan ini, maka bagian dari fluksi magnetis  $\Phi_2$  yang melalui cincin pendek tersebut, akan mempunyai fasa yang terlambat terhadap bagian lainnya yang tidak melalui cincin pendek ini. Jadi suatu efek terjadi seakan-akan bahwa kutub-kutub magnet dari inti kumparan tegangan,

<sup>7)</sup> Ibid, hal. 73.

telah bergeser pada arah perputaran dan menghasilkan suatu momen di dalam arah perputaran kepingan. Dengan mengatur posisi dari pada cincin pendek ini, maka terdapat kemungkinan untuk meniadakan pengaruh dari momen-momen gesekan.



GAMBAR 2.14<sup>8)</sup>  
PRINSIP PENGATUR BEBAN RINGAN

#### II.4.1.4. Mengelakkan Putaran Pada Beban Kosong

Bila suatu penyesuai beban ringan ditempatkan, maka terdapat kemungkinan bahwa kepingan akan berputar meskipun tidak ada beban, yaitu jika hanya kumparan tegangan yang mendapatkan energi, gejala ini disebut berputar pada beban kosong. Suatu cara untuk menghindarkan adalah dengan membuat suatu lubang pada kepingan aluminium D. Bila lubang tersebut sampai dibawah inti dari kumparan tegangan, maka jalan dari pada arus-arus putar yang terjadi di dalam kepingan akan mengalami gangguan.

Hal ini menyebabkan momen pada beban ringan akan dikurangi dan menyebabkan kepingan akan berhenti pada posisi tersebut.

<sup>8)</sup> Ibid, hal. 73.



#### II.4.2. Karakteristik KWh Meter Induksi

Karakteristik kWh meter, untuk keadaan-keadaan berbeban, pengaruh terhadap perubahan tegangan, dan pengaruh terhadap perubahan frekuensi, serta pengaruh terhadap perubahan temperatur, pada kondisi  $\cos \phi = 1$ , ataupun pada kondisi  $\cos \phi = 0.5$ ; terlihat pada tabel 2.1 berikut.

Bila karakteristik suatu kWh meter induksi diluar dari interval yang telah ditentukan, maka kesalahan penunjukkan akan cukup berarti. Untuk mengatasi hal tersebut, maka kompensasi-kompensasi untuk masing-masing keadaan ataupun kondisi tersebut harus dilakukan, dan hal ini akan sangat berpengaruh terhadap kecermatan penunjukkan meter.

TABEL 2.1<sup>9)</sup>  
PROSENTASE BATAS KESALAHAN KWH METER INDUKSI

Besar arus	Faktor Daya	Prosentase batas kesalahan dengan kelas kWh		
		0,5	1,0	2,0
Dari 10% arus nom. sampai arus maks.	1,0	±0,5	±1,0	±2,0
Pada 5% arus nominal	1,0	±1,0	±1,5	±2,5
Dari 20% arus nom. sampai arus maks.	0,5	±0,8	±1,0	±2,0
Pada 10% arus nominal	0,5	±0,5	±1,0	±2,0

<sup>9)</sup> -----, IEC Standard, Publication 521, Jenewa, Swis, 1976, hal. 39.

## TYPICAL PERFORMANCE CURVES

## LOAD CURVE

—  $V_n \cos \varphi = 1$   
 .....  $V_n \cos \varphi = 0.5$

## EFFECT OF VOLTAGE VARIATIONS

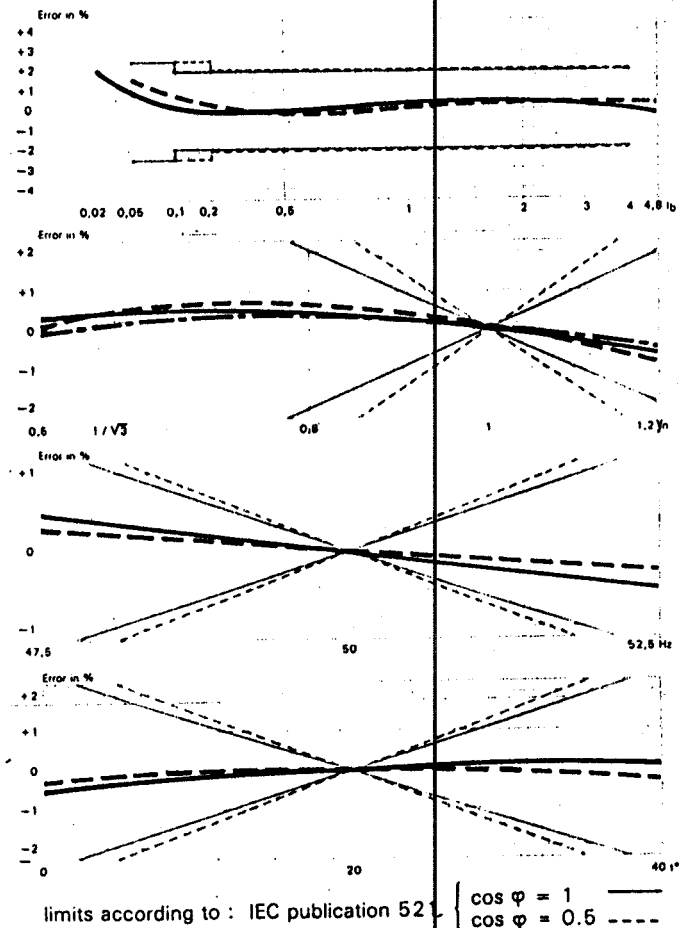
—  $I_b \cos \varphi = 1$   
 .....  $I_b \cos \varphi = 0.5$   
 .....  $0.1 I_b \cos \varphi = 1$

## EFFECT OF FREQUENCY VARIATIONS

—  $V_n I_b \cos \varphi = 1$   
 .....  $V_n I_b \cos \varphi = 0.5$

## EFFECT OF TEMPERATURE VARIATIONS

—  $V_n I_b \cos \varphi = 1$   
 .....  $V_n I_b \cos \varphi = 0.5$



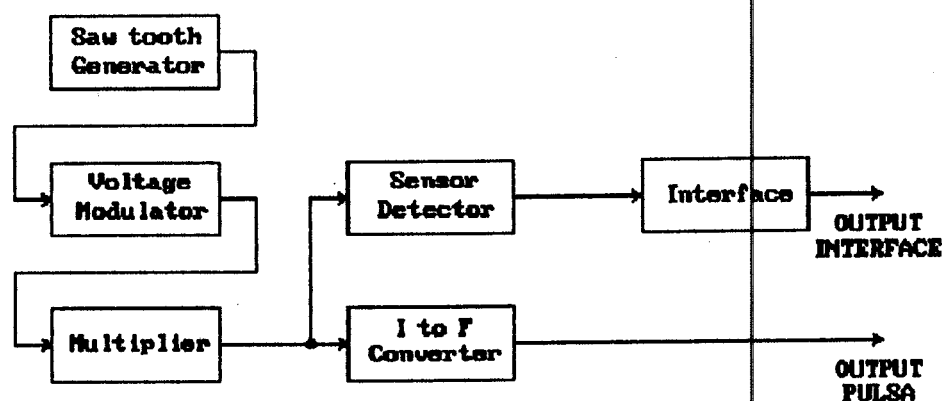
GAMBAR 2.15<sup>10)</sup>  
 KARAKTERISTIK KWH METER INDUKSI

II.4.3. Watt Hour Meter *Solid State*

Selain kWh meter induksi yang sering digunakan, sekarang telah banyak juga kWh meter *solid state*, seperti halnya kWh meter standar yang digunakan sebagai kWh meter referensi atau acuan.

Prinsip kerja kWh meter *solid state*, adalah seperti tampak pada gambar 2.16, yang prinsip kerjanya berdasarkan perkalian *MSA* (*Mark Space Amplitude*) dan integral dari dua buah gelombang tangga.

<sup>10)</sup> Ibid, hal. 39.



GAMBAR 2.16<sup>11)</sup>  
 DIAGRAM BLOK PRINSIP KERJA KWH METER *SOLID STATE*

Jika kita anggap satu fasa dari ketiga fasa, maka disini akan ada dua buah transformator yang akan menghasilkan suatu nilai arus dan tegangan dari jala-jala dengan besaran yang dibutuhkan dan dapat diterapkan pada rangkaian elektronik. Pengali *MSA* akan membentuk hasil perkalian dari arus sesaat dan tegangan sesaat dengan keluarannya berbentuk arus yang proporsional terhadap daya.

Pengubah arus ke frekuensi, mengubah arus kumulatif yang keluar dari setiap pengali menjadi frekuensi yang sebanding dengan masukkannya. Setiap pulsa itu menyatakan atau mewakili sekelompok energi.

<sup>11)</sup> Fuller, G., Mark Space Amplitude Multiplier, Schlumberger, Perancis, 1987, hal. 10.

## BAB III

### MIKROKONTROLER

#### III.1. Uraian Umum

Mikrokontroler adalah salah satu peralatan yang mempunyai peranan penting khususnya di bidang elektronika. Dapat dikatakan bahwa jantung suatu komputer adalah *Central Processor Unit (CPU)*, sedangkan suatu mikroprosesor merupakan CPU dalam kemasan chip tunggal.

Sistem pada komputer adalah CPU ditambah perangkat penunjang seperti misalnya input/output (I/O), memori, program, dan referensi waktu.

Mikrokontroler merupakan perangkat yang sangat kecil, didalamnya mengandung fungsi-fungsi yang biasa terdapat pada sistem suatu komputer. Mikrokontroler menggunakan mikroprosesor sebagai CPU, juga memori serta perangkat penunjang yang lain dalam satu kemasan chip tunggal, atau biasa disebut sebagai *single chip* atau chip tunggal. Pada mikrokontroler inipun dapat dilakukan pemrograman melalui instruksi-instruksi tertentu yang diprogramkan pada memorinya.

Dengan demikian dapat dikatakan bahwa menggunakan chip tunggal sebagai perangkat kontrol khususnya, akan jauh lebih praktis dalam segala hal dibandingkan dengan bila kita menggunakan IC logika yang konvensional.

### III.2. Chip Tunggal MC 68705 P3

Salah satu chip tunggal yang cukup handal adalah MC 68705 P3 yang merupakan keluarga 6805, produksi Motorola. Chip tunggal ini menyediakan fasilitas-fasilitas antara lain 20 jalur I/O *bidirectional* yang selaras dengan TTL/CMOS, 112 byte RAM dan 1804 byte EPROM serta fasilitas lainnya yang akan diterangkan pada subbab berikut.

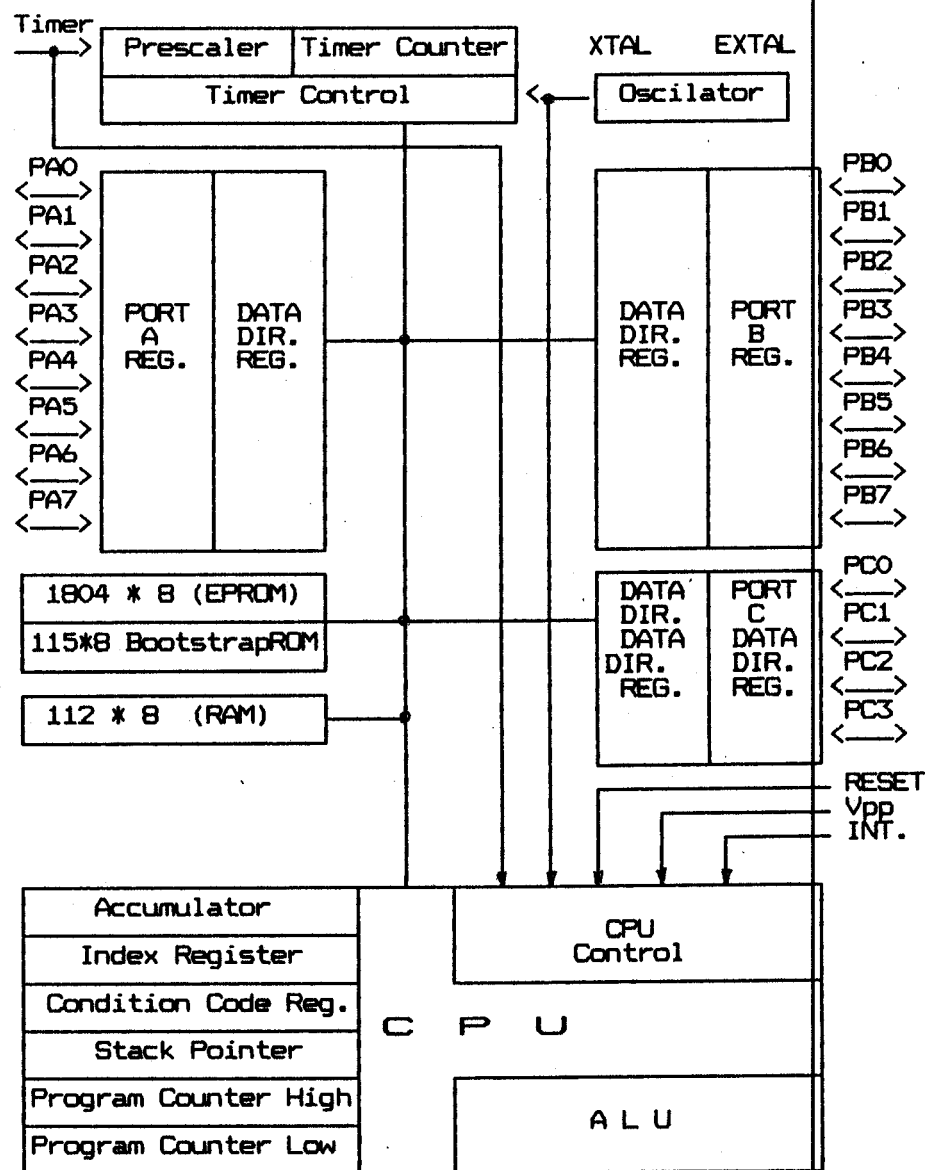
#### III.2.1. Sifat dan Karakteristik Chip Tunggal MC 68705 P3

Mikrokontroler ini termasuk salah satu dari keluarga MC 6805 yang merupakan mikroprosesor dengan 8-bit arsitektur. Secara prinsipnya mempunyai karakteristik yang sama. Perbedaan yang pokok, MC 68705 P3 ini lebih bersifat *experimentable* karena ada fasilitas EPROM.

Mikrokontroler ini memiliki 5 register pokok yang masing-masing *Accumulator (A)* dengan panjang 8-bit, *Index Register (X)* yang panjangnya 8-bit dan *Program Counter (PC)*, *Stack Pointer (SP)* yang mempunyai panjang masing-masing 11-bit, serta *Code Condition Register (CC)* dengan panjang 5-bit, seperti terlihat pada gambar 3-1. Disamping 5 register pokok, terdapat pula register-register lain yang bersifat sebagai penunjang perangkat kerasnya. MC 68705 P3 menggunakan instruksi yang memiliki efisiensi jumlah byte dan cara memprogram yang mudah, memiliki instruksi test bit dan juga bercabang, serta pengendalian interrupt serba guna.

Untuk keperluan data tabel, dapat memakai *index addressing* yang memungkinkan penempatan pada memori dimana saja yang tersedia, hal ini

karena MC 68705 P3 memiliki I/O *addressing mode* yang unik, dan juga memiliki *jump conditional* yang kompleks.



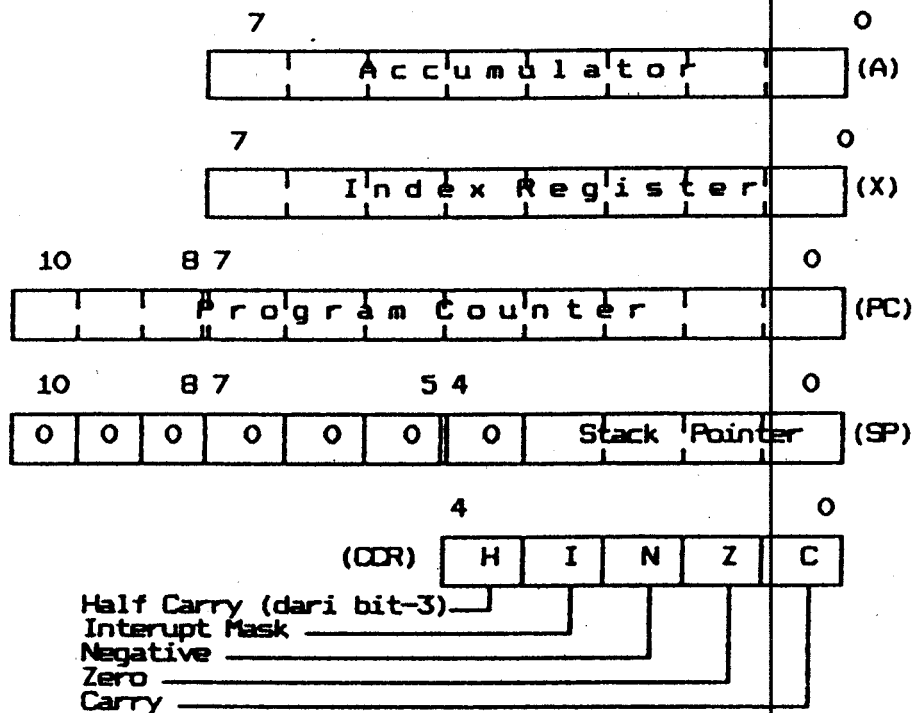
GAMBAR 3.1<sup>12)</sup>  
BLOK DIAGRAM MC 68705 P3

<sup>12)</sup> -----, Motorola Microprocessor Data Book, Motorola Inc., hal. 3-568.

Kelima register yang terdapat pada mikrokontroler adalah memori yang ada didalam mikroprosesor (bukan bagian dari memori map) seperti terlihat pada arsitektur register gambar 3.2.

### 1. Accumulator (A)

Merupakan register 8-bit serba guna yang dapat diisi data apapun yang dibangkitkan oleh program. Register 8-bit ini dapat dipakai untuk menyimpan operand, hasil dari perhitungan aritmetik ataupun manipulasi data.



GAMBAR 3.2<sup>13)</sup>  
ARSITEKTUR REGISTER MC 68705 P3

<sup>13)</sup> Ibid, hal. 3-569.

## **2. Index Register (X)**

Index Register merupakan register 8-bit yang dipakai untuk fasilitas pengalamatan memori. Dengan memberikan index dapat dipakai untuk mengakses blok-blok data dalam memori menggunakan sebuah instruksi. Index Register ini berisi harga 8-bit yang dijumlahkan dengan harga yang ada pada operand untuk instruksi baca, modifikasi atau tulis. Disamping itu index register dapat juga dipakai sebagai tempat penyimpanan sementara.

## **3. Program Counter (PC)**

Program Counter berisi alamat dari instruksi berikutnya atau operand instruksi yang diambil oleh prosesor. Pada umumnya PC memberikan satu lokasi memori untuk sesaat sebagai instruksi dan operand instruksi dijalankan. Sedangkan pelayanan interrupt, jump, dan percabangan mengakibatkan PC diisi dengan alamat memori yang berbeda dengan lokasi pengurutan normal berikutnya. PC mempunyai panjang 11-bit.

## **4. Stack Pointer (SP)**

Merupakan register 11-bit yang berisi alamat dari lokasi yang kosong/bebas pada stack. Penunjukan Stack berada pada posisi 07F<sub>H</sub> selama kondisi reset dari MCU (*Microcontroller Unit*) atau instruksi RSP (*Reset Stack Pointter*) yang berarti mereset kondisi penunjukkan stack. Penunjukan stack akan *decrement* setiap ada data yang di tumpuk pada stack dan *increment* setiap ada data yang diambil. Saat mengakses memori, enam (6) bit dari



MSB telah set pada 000011 secara tetap. Keenam bit ini disertakan pada lima (5) bit dari LSB untuk menyediakan alamat dengan range  $10_H$  sampai  $07F_H$ . Pengambilan routine dibatasi untuk tidak lebih dari 15 tingkat karena jumlah lokasi RAM yang disediakan/diperbolehkan hanya 31 byte.

### 5. *Condition Code Register (CC)*

Register ini berisi lima petunjuk status sebagai gambaran dari hasil perhitungan aritmetik dan operasi lain dari MCU. Kelima bit ini adalah *half-carry (H)*, *interrupt (I)*, *negative (N)*, *zero (Z)*, dan *carry/borrow (C)*.

Untuk mengetahui fungsinya dapat dilakukan test secara individu dengan memberikan instruksi tertentu dengan hasil sesuai dengan kondisi masing-masing bit tersebut diatas.

#### a. *Half-carry bit*

Kondisi ini dipakai untuk operasi aritmetik *BCD (binary code decimal)* sebagai akibat dari instruksi ADC atau ADD. Bit-H ini akan dalam kondisi "1" saat terjadi carry antara bit-3 dan bit-4.

#### b. *Interrupt (I)*

Saat timer dan external interrupt tidak dibolehkan (*disable*) maka bit-H dalam kondisi set "1". Meskipun demikian, jika bit-H dalam kondisi set "1" terjadi permintaan pelayanan external interrupt akan ditahan dan kemudian diproses setelah bit-H kembali reset "0".

#### c. *Negative (N)*

Bit-N ini akan set "1" saat hasil akhir operasi aritmetik, logika atau

data manipulasi negatif. Dalam artian bit-7 dari hasil berlogika "1".

d. *Zero (Z)*

Bit-Z akan set 1 saat hasil akhir operasi aritmetik, logika dan bit manipulasi berlogika "0".

e. *Carry/Borrow (C)*

Bit-C digunakan untuk menunjukkan ada tidaknya carry dari hasil penjumlahan atau *borrow* dari hasil pengurangan. Hal inipun terjadi pada pemakaian instruksi percabangan atau pergeseran dan perputaran.

### III.2.2. Memori Map MC 68705 P3

MCU mempunyai kemampuan pengalamatan sampai 2048 byte dari memori dan register I/O. Seperti terlihat pada gambar 3.3.

EPROM untuk pemakai pada alamat \$080 - \$783, ROM untuk program *bootstrap*, RAM yang menempati 112 byte memori dengan 31 byte dipakai untuk stack, *Mask Option Register (MOR)*, *Program Control Register* dan I/O. Sedangkan vektor interrupt yang merupakan EPROM cadangan terletak pada alamat \$7F8 - \$7FF.

000	I/O Port Timer & RAM(128 bytes)	\$000	0	Port A	\$000
127		\$07F	1	Port B	\$001
128	Page Zero User EPROM (128 bytes)	\$080	2	1 1 1 1 Port C	\$002
255		\$0FF	3	Tidak dipakai	\$003
256		\$100	4	DDR Port A	\$004
	Main User EPROM (1668 bytes)		5	DDR Port B	\$005
			6	1 1 1 1 DDR P C	\$006
			7	Tidak dipakai	\$007
			8	Timer Data Reg.	\$008
1923		\$783	9	Timer Cont.Reg.	\$009
1924	Mask Opt. Reg.	\$784	10	Tidak dipakai	\$00A
1925		\$785	11	Programming Control Reg.	\$00B
	Bootstrap ROM (115 bytes)	\$7F7	12		\$00C
	Timer Interrupt	\$7FB	15	Tidak dipakai	\$00F
	Ekt. Interrupt		16		\$010
	S W I			RAM (112 bytes)	
	Reset			31 bytes untuk Stack	
2047		\$7FF	128		\$07F

GAMBAR 3.3<sup>14)</sup>  
MEMORI MAP MC 68705 P3

### III.2.3. Perangkat Lunak Chip Tunggal MC 68705 P3

Pada prinsipnya perangkat lunak pada MC 68705 P3 tidak lepas dari pendahulunya yaitu mikroprosesor MC 6800. Disamping pemrogramannya yang mudah, instruksi yang digunakan mempunyai efisiensi jumlah byte, serta memiliki instruksi test dan percabangan. Pada dasarnya MC 68705 P3 mempunyai 59 instruksi dasar yang dibagi menjadi 5 jenis yang berbeda

<sup>14)</sup> Ibid, hal 3-571.

yaitu:

1. Instruksi register/memori
2. Instruksi baca - modifikasi - tulis
3. Instruksi percabangan
4. Instruksi pengontrolan
5. *Bit manipulation*

Instruksi register/memori kebanyakan menggunakan dua operand yaitu accumulator atau index register dan yang kedua menggunakan memori dengan mode pengalamatan.

Instruksi baca - modifikasi - tulis dilakukan untuk membaca memori atau register, memodifikasi atau melakukan test, yang kemudian menuliskan kembali hasil modifikasi ke memori atau register.

Pada Instruksi percabangan, MC 68705 P3 mempunyai banyak variasi percabangan dengan mengacu pada kondisi tertentu. Instruksi percabangan ini menduduki 2 byte instruksi.

Instruksi pengontrolan merupakan instruksi pengontrol prosesor dalam melaksanakan program.

*Bit manipulation* merupakan kemampuan MC 68705 P3 dalam manipulasi bit pada lokasi tertentu untuk lebih memudahkan pemrograman. Kemampuan tersebut untuk men-set atau men-clear setiap bit pada 256 lokasi memori awal dimana terdapat port register, port DDR, timer control dan RAM. Disamping itu sebagai tambahan dapat juga dilakukan test dan percabangan pada setiap posisi bit didalam area 256 byte awal.



Mode pengalamatan pada MC 68705 P3 mempunyai beberapa jenis yang berbeda dan dapat memberi kemudahan pada pemrograman, untuk mengoptimasi kode-kode pada semua situasi.

Beberapa mode pengalamatan yang ada antara lain adalah mode pengalamatan berindex yang memberikan kemungkinan menempatkan tabel data, tabel kode konversi serta tabel skala dimanapun pada memori.

Dua byte pengalamatan langsung (*direct addressing*) pada mode pengalamatan langsung yang dapat menjangkau RAM, I/O, dan 128 byte EPROM. Pada mode ini *efektif address* adalah data byte yang ada di operand.

Pada mode pengalamatan *extended* memungkinkan intruksi jump yang mencakup seluruh memori. Dengan menggunakan bahasa *assembler* dari Motorola akan secara otomatis memilih mode *extended* atau *direct*.

Adapun mode pengalamatan yang lain secara garis besar adalah : *immediate, relative, bit set/clear, bit test*, dan percabangan serta *inherent*.

#### III.2.4. Perangkat Keras Chip Tunggal MC 68705 P3

Sebagai 8-bit NMOS mikrokomputer chip tunggal, MC 68705 P3 yang merupakan pendahulunya. Sesuai dengan gambar 3.1, terlihat bahwa pada chip tunggal MC 68705 P3 terdapat 3 *bidirectional port*, yang masing-masing 8-line port A, 8-line port B, dan 4-line port C. Disamping itu terdapat pula interrupt line, timer, sistem kontrol. Secara internal chip tunggal MC 68705 P3 mempunyai ROM, EPROM, pembangkit clock dan 8-bit timer de-

ngan *prescaler* serta adanya *bootstrap* yang dipakai untuk memprogram diri sendiri.

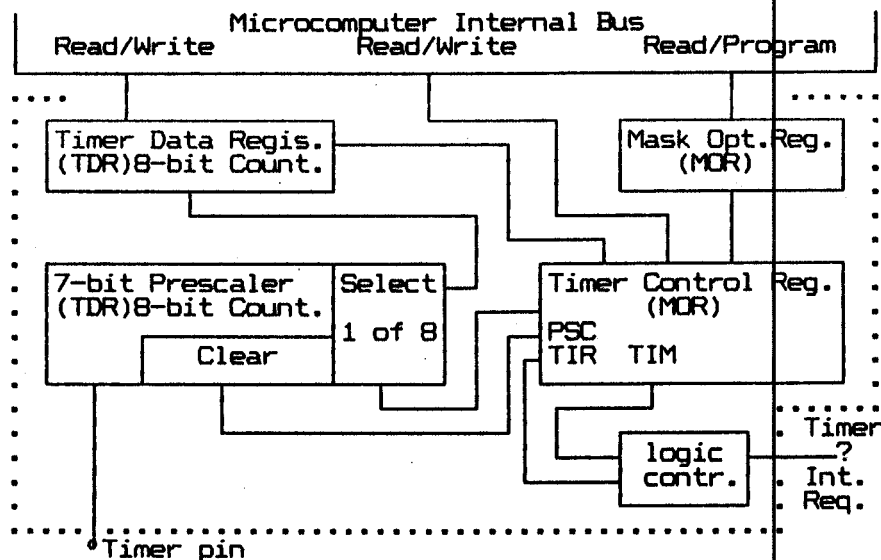
### 1. *Timer*

Timer disini merupakan satu fasilitas yang disediakan oleh MC 68705 P3 dengan memberikan counter 8-bit yang dikendalikan oleh 7-bit *prescaler* yang dapat diprogram secara perangkat lunak, dengan demikian akan kita dapatkan berbagai sumber timer melalui *Timer Control Register (TCR)* atau *Mask Option Register (MOR)* seperti terlihat pada gambar 3.4.

Counter 8-bit tersebut dapat diambil dengan pengontrolan program dan kemudian berkurang (*decrement*) menuju nol. Selanjutnya bila telah mencapai nol akan mengakibatkan kondisi set pada *Timer Interrupt Request (TIR)* yaitu bit ke-7 dari TCR.

b7	b6	b5	b4	b3	b2	b1	b0
TIR	TIM	TIN	TIE	PSC	PS2	PS1	PS0

Untuk menghalangi interrupt dari timer, dapat digunakan *Timer Interrupt Mask (TIM)* dari TCR yang merupakan bit ke-6. Saat bit 1 dari CC register dan bit TIM pada kondisi clear, prosesor menerima interrupt. Untuk menanggapi interrupt ini, pertama-tama menyimpan langkah yang sedang diambil ke dalam stack kemudian mengambil vektor interrupt dari timer pada lokasi \$7F8 dan \$7F9 yang selanjutnya me-ngerjakan (*execute*) routine interrupt.



GAMBAR 3.4<sup>15)</sup>  
BLOK DIAGRAM TIMER

TIR dapat dikembalikan ke kondisi clear melalui pengontrolan perangkat lunak, *master reset* atau *external reset*. Jika interrupt dihalangi maka dengan instruksi BCLR, dapat dilakukan tanpa mengabaikan sinyal interrupt, sehingga TIR di-clear pada saat melaksanakan routine interrupt. Pada blok diagram dapat dilihat hubungan antara TIR dan TIM pada TCR serta hasil permintaan interrupt itu sendiri. Perhitungan *counter* menuju \$00 akan berlanjut kembali ke \$FF yang selanjutnya *decrement* lagi. *Counter* ini dapat dibaca sewaktu-waktu oleh MCU melalui *Timer Data Register (TDR)*. Hal ini dapat dipakai oleh program untuk mengetahui selang waktu mulai dari pelayanan interrupt dari timer tanpa mengganggu proses perhitungan. Dalam hal ini TDR tidak dipengaruhi oleh reset.

<sup>15)</sup> Ibid, hal. 3-574.

Sesuai dengan blok diagram, *prescaler* pada TCR atau MOR option memilih satu diantara 8 bagian pada pembagi 7 dimana bagian ke-8 adalah *bypass*.

Pada kondisi reset *prescaler* dan counter diinisialisasikan pada kondisi set, sedang TIR yang merupakan bit ke-7 pada TCR dalam kondisi clear dan TIM pada kondisi set. Untuk TCR bit 0 sampai bit 5 disesuaikan dengan MOR pada kondisi set, yang pada saat setelah keadaan reset dapat diseleksi.

*Prescaler* disini merupakan pembagi 7 bit yang digunakan untuk memberikan panjang maksimum dari timer. *Prescaler* dapat diclearkan saat bit ke-4 (b3) dari TCR yaitu *prescaler clear (PSC)* dalam kondisi set. Meskipun demikian PSC bit yang hanya untuk ditulis (*write only bit*) yang kalau dibaca selalu menunjukkan logika "0".

Sesuai dengan blok diagram timer, terdapat dua pelayanan timer kontrol yaitu :

1. Pengontrolan perangkat lunak
2. Pengontrolan MOR

Melalui pengontrolan perangkat lunak, input *prescaler* dapat disusun menjadi tiga mode yang berbeda ditambah satu mode disable. Hal ini tergantung pada nilai yang di-tulis pada TCR bit 4 dan 5 (TIE, TIN). Seluruh bit pada TCR dapat berupa baca-tulis kecuali bit 3 yang hanya merupakan bit untuk ditulis saja seperti penjelasan didepan.

*Timer Option (TOPT)* yang merupakan bit 6 dari MOR diprogram pada kondisi logika "0" untuk pengontrolan mode melalui perangkat lunak.



Mode 1 untuk input timer dipakai saat TIE dan TIN diprogram pada kondisi logika "0". Input timer yang dipakai berupa internal clock ( $\Phi 2$ ) sedangkan timer input pin diabaikan. Mode dengan clock secara internal ini dapat digunakan untuk pembangkitan interrupt secara periodik yang dapat dipakai sebagai referensi frekuensi.

TIE=1 dan TIN=0 untuk mode 2. Clock internal dan sinyal timer input di-AND-kan untuk membentuk timer input (fpin) pada masukan prescaler. Mode ini dapat dipakai untuk mengukur lebar pulsa eksternal dengan akurasi perhitungan sampai  $\pm 1$ .

Pada mode 3 TIE=0 dan TIN=1, yang berarti tidak ada input frekuensi prescaler dan timer diabaikan.

TIE dan TIN untuk mode 4 sama dengan 1, yang berarti input timer dibentuk oleh clock dari luar. Clock dari luar ini dapat digunakan untuk perhitungan luar seperti misalnya membangkitkan frekuensi luar untuk pelayanan interrupt secara periodik.

Pengontrolan MOR menggunakan bit 6, 7 pada TCR sebagai bit baca tulis. Bit 3 hanya sebagai bit baca, sedangkan 5 bit yang tersisa tidak mengakibatkan sesuatu pada operasi baca tulis. Mode dengan pengontrolan MOR didapat dengan memprogram TOPT dalam MOR pada logika "1". Level logika dari TCR pada bit 0, 1, 2, dan 5 harus didefinisikan saat memprogram EPROM, sedangkan TIE, dan PSC di-set pada logika "1".

Adapun fungsi masing-masing register yang tampak pada blok diagram timer akan dijelaskan sebagai berikut:

### A. Timer Control Register (TCR)

TCR disini menempati alamat \$009 pada memori map, serta merupakan register 8-bit. TCR mempunyai fungsi antara lain untuk konfigurasi mode operasi, menentukan rasio dari prescaler serta membangkitkan permintaan sinyal interrupt timer. Konfigurasi dari TCR ditentukan oleh bit-6 (*TOPT - Timer mask/programable Option*) didalam MOR. Dengan demikian untuk kondisi  $TOPT=1$ , TCR dapat ditetapkan oleh pemakai melalui MOR yang dalam hal ini sesuai dengan emulasi dari MC 6805 P2/P6.

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	1	1	1	1	1	1	alamat \$009

TCR dengan  $TOPT-MOR = 1$

b7	b6	b5	b4	b3	b2	b1	b0	
TIR	TIM	TIN	TIE	PSC	PS2	PS1	PS0	alamat \$009

TCR dengan  $TOPT-MOR = 0$

Sedangkan pada  $TOPT-MOR = 0$ , TCR digunakan sebagai pengontrol timer secara perangkat lunak dengan penjelasan masing-masing bit sebagai berikut :

#### a. TIR *Timer Interrupt Request*

Dipakai untuk mengenali adanya interrupt saat berlogika "1".

1 = set saat TDR berubah pada harga "0"

0 = kondisi clear dengan cara reset eksternal, power-ON reset atau dibawah pengontrolan perangkat lunak.

**b. TIM *Timer Interrupt Mask***

Digunakan untuk menghalangi interrupt timer saat berlogika "1", sedangkan saat berlogika "0" untuk sebaliknya.

**c. TIE *Timer Input Select***

Digunakan untuk memilih sumber clock input yang dalam hal ini berlogika "1" untuk pemilihan clock eksternal (pin-7), dan berlogika "0" untuk clock internal ( $\Phi 2$ ).

**d. TIE *Timer External Input Enable***

Digunakan untuk pemakaian pin timer eksternal yang digabungkan dengan TIN seperti tabel 3.1.

**e. PSC *Prescaler Clear***

Merupakan bit tulis saja dengan ketentuan menuliskan "1" untuk mereset prescaler menjadi pada kondisi logika "0".

**f. PS0, PS1, PS2 *Prescaler Select bit***

Digunakan untuk memilih satu dari 8 output prescaler seperti terlihat pada tabel 3.2.

TABEL 3.1<sup>16)</sup>  
HUBUNGAN TIN & TIE

TIN	TIE	KETERANGAN
0	0	Internal clock
0	1	Internal and External
1	0	No clock
1	1	External clock

<sup>16)</sup> Ibid, hal 3-575.

TABEL 3.2<sup>17)</sup>  
 PRESCALER SELECT BIT

PS2	PS1	PS0	KETERANGAN
0	0	0	1 (tanpa prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	62
1	1	1	128

### B. Mask Option Register (MOR)

MOR diimplementasikan dalam EPROM pada saat menggunakan emulasi untuk MC 6805 P2/P6. Dalam hal ini MOR ber-isinya semua kondisi awal untuk kepentingan pemrograman. Adapun konfigurasi MOR dapat diterangkan sebagai berikut:

b7	b6	b5	b4	b3	b2	b1	b0
CLK	TOPT	CLS	TIE		P2	P1	P0

Bit 7 (CLK -clock) dipakai untuk memilih jenis oscillator clock, dimana "1", untuk jenis RC dan "0" untuk pemakaian kristal.

TOPT merupakan timer option yang digunakan untuk menentukan konfigurasi TCR seperti yang sudah dijelaskan terlebih dahulu. Kondisi "1" pada TOPT merupakan timer/prescaler dari MC 6805 P2/P6, dimana seluruh bit tidak dapat dilihat oleh pemakai kecuali bit 6 dan 7 saja. Sedangkan bit 5, 2, 1,

<sup>17)</sup> Ibid, hal. 3-575.

dan 0 dari MOR ditentukan sama dengan MC 6805 P2/P6. Untuk TOPT pada kondisi "0" diimplementasikan dengan pemrograman secara perangkat lunak.

CLS - *Timer/prescaler Clock Source* dipakai untuk pemilihan clock. Kondisi "1" untuk pemakaian eksternal clock melalui pin 7 timer dan kondisi "0" untuk internal clock.

TIE tidak dipakai jika TOPT pada kondisi "1". Pada TOPT="0" merupakan setting nilai TIE pada TCR.

P0, P1, P2, seperti halnya PS0, PS1, PS2, pada TCR yang berfungsi untuk memilih satu dari 8 output prescaler yang disediakan.

## 2. Input/Output

Fasilitas Input maupun Output pada MC 68705 P3 terdiri dari 20-line yang dibagi menjadi 3 bagian yaitu : 8-bit untuk port A, 8-bit untuk port B, serta 4-bit untuk port C. Masing-masing line pada port dapat diprogram secara perangkat lunak untuk difungsikan sebagai input atau output sesuai dengan isi dari *Data Direction Register (DDR)* port yang ditentukan.

DDR disini merupakan register 8-bit yang hanya bersifat tulis yang mempunyai ketentuan, jika DDR port berlogika "1", maka akan bertindak sebagai output, dan berlogika "0", akan bertindak sebagai input, seperti pada tabel 3.3 berikut.

Pada keadaan setelah reset, DDR diinisialisasikan dengan logika "0", maka akan bertindak sebagai input. Seluruh pin input/output tersebut bersifat kompatibel terhadap TTL dengan perkecualian untuk port A saat

bertindak output bersifat kompatibel terhadap CMOS dan port B dan port C saat bertindak sebagai input bersifat kompatibel terhadap CMOS.

TABEL 3.3<sup>10)</sup>  
FUNGSI PIN I/O

Bit Data Direction Register	Bit Output Data	Kondisi Output	Input Ke MCU
1	0	0	0
1	1	1	1
0	X	Hi - Z	Pin

### 3. Interrupt

Pelayanan interrupt pada MC 68705 P3 mempunyai tiga cara yang berbeda yaitu :

- a. Melalui input interrupt (INT).
- b. Melalui permintaan interrupt timer secara internal.
- c. Memakai perangkat lunak dengan instruksi interrupt (*SWI - Software interrupt Instruction*).

Saat ada permintaan pelayanan interrupt terjadi, MCU mendorong register ke dalam stack serta bit-I pada CC re-gister set untuk mencegah terjadinya permintaan pelayanan interrupt tambahan. Kemudian setelah alamat *routine interrupt* diberikan oleh vektor alamat, MCU menjalankan routine interrupt. Setelah routine interrupt selesai (sampai pada instruksi RTI - *Return from Interrupt*), MCU mengambil register dari stack dan kembali menjalankan proses yang tertunda. Perlu diketahui bahwa penumpukan register, setting bit-

<sup>10)</sup> Ibid, hal. 3-570.

I pada CC register dan pengambilan vektor alamat membutuhkan  $11 t_{\text{cycle}}$  periode. Permintaan pelayanan interrupt dapat dibedakan atas prioritasnya seperti pada tabel 3.4 berikut.

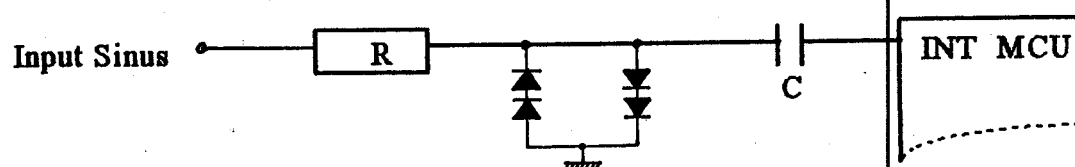
TABEL 3.4<sup>19)</sup>  
PRIORITAS INTERRUPT

INTERRUPT	PRIORITAS	ALAMAT VEKTOR
RESET	1	\$7FE - \$7FF
SWI	2 (lihat uraian)	\$7FC - \$7FD
INT	3	\$7FA - \$7FB
TIMER	4	\$7F8 - \$7F9

Prioritas pertama adalah pada reset vektor interrupt yang perlu diperhatikan saat melakukan pemrograman EPROM.

Prioritas kedua berlaku hanya saat bit-I dari CC register dalam kondisi set. Tetapi pada saat bit-I dari CC register pada kondisi clear, SWI menduduki prioritas ke-4, sedangkan INT ke-2 serta timer menduduki prioritas ke-3 dari keseluruhan permintaan pelayanan interrupt.

Secara internal, interrupt dari luar disinkronisasikan dan ditahan pada transisi turun INT. Sedangkan sinyal input berbentuk sinus dapat dipakai untuk membangkitkan eksternal interrupt seperti yang terlihat pada gambar 3.5 berikut, dengan menggunakan *zero crossing detector*.



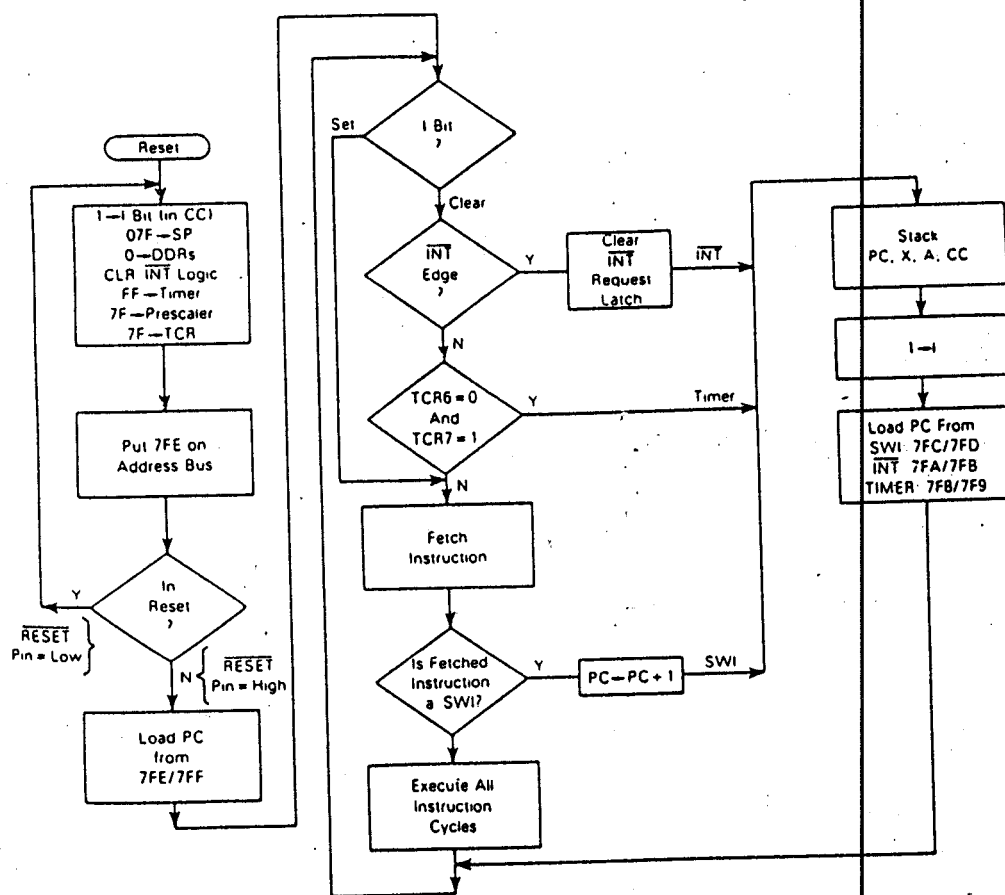
GAMBAR 3.5<sup>20)</sup>  
RANGKAIAN EKSTERNAL INTERRUPT

<sup>19)</sup> Ibid, hal. 3-572.

<sup>20)</sup> Ibid, hal. 3-574.

Permintaan pelayanan interupt juga dapat dilakukan dalam bentuk perangkat lunak (SWI), hal ini dapat dijalankan tanpa memperhatikan keadaan dari bit-I CC register. Pada umumnya SWI dipakai sebagai *break point* dari *debugging* atau sistem *call*.

Adapun cara kerja pelayanan interupt tersebut diatas dapat digambarkan pada diagram alir gambar 3.6.



GAMBAR 3.6<sup>21)</sup>  
DIAGRAM ALIR PELAYANAN INTERRUPT

<sup>21)</sup> Ibid, hal. 3-573.



#### 4. Bootstrap Program

MC 68705 P3 dilengkapi dengan program untuk memprogram EPROM yang menjadi salah satu fasilitasnya. Secara garis besar pemrograman EPROM tersebut dilakukan dengan cara mentransfer data pada eksternal EPROM yang telah diprogram terlebih dahulu dengan menggunakan rangkaian yang sangat sederhana. Selanjutnya untuk penjelasan yang lebih terperinci diterangkan pada sub-bab Teknik Pemrograman.

#### III.2.5. Teknik Pemrograman

Seperti yang telah dijelaskan didepan, MC 68705 P3 mempunyai program *bootstrap* pada ROM yang dipakai untuk memprogram dirinya sendiri. Dalam hal ini, terdapat *Program Control Register (PCR)* yang merupakan register 8-bit yang menyediakan bit-bit pengontrol pemrograman EPROM.

PCR dimanfaatkan pada 3 bit LSB, sedangkan 5 bit MSB berada pada kondisi logika "1".

b7	b6	b5	b4	b3	b2	b1	b0
1	1	1	1	1	VPON	PGE	PLE

Sesuai dengan keterangan diatas, bit 0 - *Program Latch Enable (PLE)* akan set pada waktu reset dan dapat di-clear-kan setiap saat. Pada saat clear akan menahan alamat dan data kedalam EPROM dan pada kondisi set, data dapat dibaca dari EPROM.

*Program Enable (PGE)*, merupakan bit yang hanya dapat dibaca. Pada kondisi "0", menunjukkan bahwa tegangan tinggi ada pada Vpp dan kondisi "1", berarti sebaliknya, serta memutus hubungan PGE dan PLE dari chip. Sebagai tambahan, VPON sebenarnya tidak untuk menunjukkan bahwa level tegangan telah sesuai untuk pemrograman, melainkan digunakan sebagai pengaman pengunci untuk pemakai saat beroperasi pada mode normal. Secara keseluruhan dapat dilihat pada tabel 3.5, berikut.

Untuk memprogram diri sendiri MC 68705 P3 mempunyai 115 bit mask ROM, yang berisi program *bootstrap* yang berada pada alamat \$785 sampai \$7F7, dan start program pada vektor alamat dengan lokasi \$7F6 dan \$7F7. Vektor ini diambil pada saat VIHTP diumpankan pada pin 7 (*timer/boot*). Pemrograman dimulai saat reset set, yang kemudian membaca isi master EPROM berupa EPROM eksternal dengan isi program yang diinginkan.

TABEL 3.5<sup>22)</sup>  
PROGRAMMING CONTROL REGISTER

VPON	PGE	PLE	KONDISI PEMROGRAMAN
0	0	0	Mode Pemrograman
1	0	0	PGE & PLE diabaikan
0	1	0	Alamat & Data Latch
1	1	0	PGE & PLE diabaikan
0	0	1	Kondisi Invalid PGE = 0 -
1	0	1	jika PLE = 0
0	1	1	Tegangan tinggi pada Vpp
1	1	1	Mode operasi

<sup>22)</sup> Ibid, hal. 3-576.

### III.3. Program Simulasi 6805 AVSIM

Untuk mendukung mikrokontroler MC 68705 P3, disediakan perangkat lunak berisi program simulasi produksi *AVOCET System, Inc. Software*, ini dapat dipakai untuk simulasi keluarga Motorola 6805, yang berdasarkan jenis ROM-nya meliputi HMOS ROM, CMOS ROM, NMOS EPROM, dan EPROM.

Perangkat keras yang dipakai minimum komputer IBM. PC/AT dengan sistem operasi PC-DOS atau MS-DOS. Pada dasarnya perangkat lunak ini merupakan pendamping *Cross Assembler XASM05* dalam membuat program untuk diisikan ke dalam PROM atau EPROM. Dengan simulasi perangkat lunak, pemakai dapat melakukan *debugging* pada program secara langkah demi langkah (*single step*) atau dengan *Break Point*.

Fasilitas yang disediakan perangkat lunak ini di-samping berbagai jenis keluarga Motorola 6805, juga ditampilkan CPU secara visual lengkap dengan *register, flag, stack, status*, daerah memori, serta program *disassembled*, yang dapat diubah saat simulasi dijalankan. Disamping itu dapat dilakukan interaksi langsung dengan pin-pin eksternal oleh pemakai seperti misalnya memasukkan data melalui pin input atau output.

Langkah-langkah yang dilakukan untuk memakai simulasi perangkat lunak ini, pertama kali dilakukan penyuntingan teks program dengan bahasa assembly, melalui pe-nyunting teks yang umum dipakai. Kemudian teks program yang disunting tersebut di-*crossassembler*-kan, sehingga menghasilkan file *object code* dengan ekstensi .HEX.

Cara lain yang dapat dilakukan adalah dengan cara menyunting langsung program yang dimaksud kesimulatornya (6805 AVSIM).

### III.3.1. Pelaksanaan Simulasi dan Perintah pada Simulator.

Dalam pelaksanaannya Software Simulasi keluarga Motorola 6805 pilihan mikrokontroler yang disediakan. Pada saat awal ini AVSIM belum mempunyai ROM pada memori yang *preallocated* kecuali kita menggunakan *object module* hasil *cross assembler*. Dengan demikian harus ditentukan terlebih dahulu menggunakan perintah *view memory-map* untuk melihat peta memori dan *set memory-map* untuk menambah ROM. Perintah tersebut terlihat pada menu tampilan bagian bawah.

AVSIM mempunyai dua mode yang digunakan untuk melakukan interaksi, yaitu *Command Mode* dan *Display Mode*. Pada mode *display*, pemakai dapat melakukan penyuntingan register, memori, register part, dan pin interrupt dengan cara meletakkan kursor pada bagian yang akan disunting, kemudian memberikan harga yang diinginkan. Hal ini tetap dapat dilaksanakan meskipun simulator sedang menjalankan simulasinya.

Sedangkan pada mode perintah, pemakai dapat melaksanakan perintah-perintah yang telah disediakan di bagian bawah tampilan. Perintah tersebut secara garis besar meliputi *load*, untuk mengambil program, *set*, *reset*, *view*, serta *dump*, yang digunakan untuk melihat memori pada daerah tertentu pada alamat *absolut* atau *indirect*. Perintah *patch* untuk penyuntingan langsung, *execute*, *expression*, digunakan untuk memberikan ekspresi

ditempat kursor, *memory*, serta *help*.

Perintah penunjang lainnya antara lain +, -, untuk *increment* dan *decrement*, serta fungsi-fungsi perintah sebagai berikut :

F1 = menjalankan simulator (bersifat *toggle*).

F2 = menaikkan letak kursor *break point*.

F3 = men-set kursor *break point*.

F4 = menurunkan kursor *break point*.

F5 = memilih kecepatan simulasi.

F6 = *display cycle* (bersifat *toggle*).

F7 = memilih mode bilangan.

F8 = *skip* (bersifat *toggle*).

F9 = membatalkan satu langkah instruksi.

F10 = menjalankan satu langkah instruksi.

Alt-F6 = memilih simulasi untuk tiap langkah (bersifat *toggle*).

## BAB IV

### PERENCANAAN DAN PEMBUATAN

Dalam penggunaan energi listrik pemakaiannya perlu dicatat dengan menggunakan kWh meter, sehingga akan diketahui berapa jumlah pemakaian dalam jangka waktu tertentu. Mengingat akan pentingnya hal ini, maka diperlukan kWh meter yang ketelitiannya memenuhi standar yang telah ditentukan. Untuk melakukan pengecekan ketelitian kWh meter, perlu mengetahui besarnya konstanta (k) yang dimiliki oleh kWh meter induksi yang sedang di tera dan konstanta (C) kWh meter standar. Besarnya satuan kWh meter induksi adalah putaran/kWh, sedang besarnya satuan kWh meter yang dipakai dalam perencanaan ini adalah 10  $\mu$ Wh.C/pulsa (buatan ENERTEC dengan tipe E4Y21E).

Pengecekan ketelitian kWh meter yang biasanya dilakukan adalah dengan manual, yaitu dengan cara membandingkan jumlah pulsa yang ditampilkan oleh kWh meter standar dengan jumlah pulsa dari hasil perhitungan, pada n kali putaran kWh meter induksi. Untuk menghitung jumlah pulsa pada n kali putaran adalah :

$$\text{jumlah pulsa} = \frac{n}{kC}$$

Misalkan konstanta kWh meter induksi (buatan Mecoindo dengan tipe M2X) adalah 1250 putaran/kWh dan konstanta kWh meter adalah 11 (pada tegangan 220 volt, arus 5 ampere), dengan 1 kali putaran, maka :

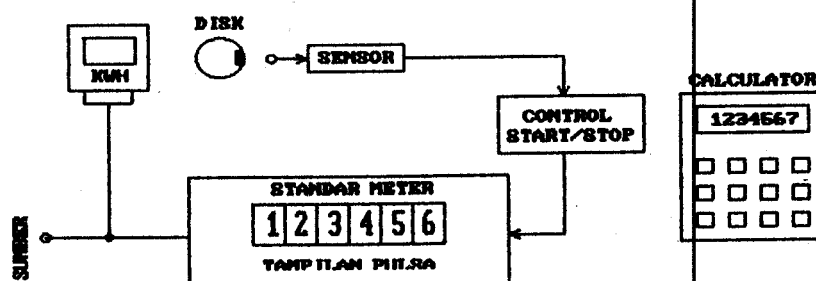
$$\begin{aligned}
 \text{jumlah pulsa} &= \frac{1}{k C} \\
 &= \frac{1}{1250 \text{ putaran/kWh} \cdot 11 \cdot 10^{-5} \text{ Wh/pulsa}} \\
 &= \frac{10^8}{1250 \cdot 11} \left[ \frac{\text{pulsa}}{\text{putaran}} \right] \\
 &= 72727 \left[ \frac{\text{pulsa}}{\text{putaran}} \right]
 \end{aligned}$$

misalnya pulsa hasil perhitungan = X, sedang tampilan kWh meter standar menunjukkan jumlah pulsa = Y, maka :

$$\text{prosentase kesalahan} = \frac{X - Y}{Y} \times 100 \%$$

Bila prosentase kesalahan sama dengan negatif, maka kWh meter tersebut meter berputar pelan, tetapi bila prosentase kesalahan sama dengan positif, maka kWh meter berputar cepat. Kemudian dilakukan peneraan pada kWh meter induksi tersebut.

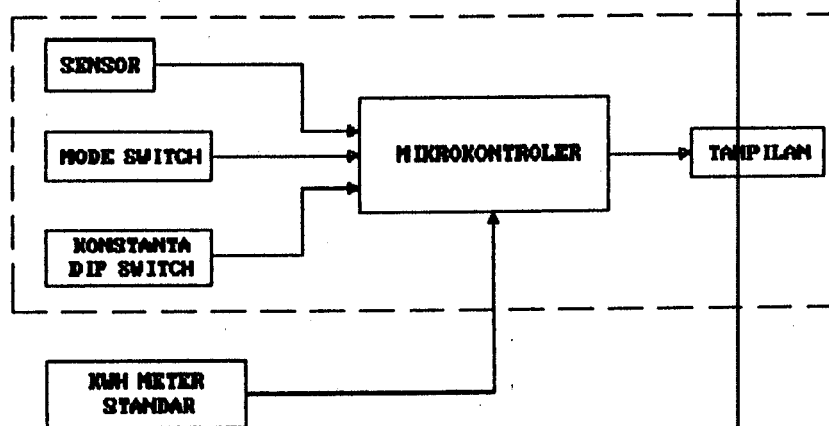
Perhitungan-perhitungan seperti diatas sering dilakukan setiap kali melakukan pengetesan kWh meter induksi pada satu kondisi, dengan demikian akan mengurangi efektifitas kerja. Untuk itu perlu alat penghitung kesalahan kWh meter yang dapat menampilkan langsung prosentase kesalahan.



GAMBAR 4.1  
PENERAAN KWH METER INDUKSI SECARA MANUAL

#### IV.1. Blok Perencanaan

Dalam perencanaan dan pembuatan alat penghitung kesalahan kWh meter induksi, yang diimplementasikan dengan chip tunggal MC 68705 P3, terlebih dahulu dilakukan perencanaan blok diagram, seperti terlihat pada gambar 4.2.



GAMBAR 4.2  
BLOK DIAGRAM PENGHITUNG KESALAHAN KWH METER

Pada blok diagram diatas, terdapat 5 bagian pokok, yaitu :

1. Sensor.
2. Mode Switch.
3. Konstanta DIP Switch.
4. Mikrokontroler.
5. Tampilan.

##### 1. Sensor.

Sensor disini berfungsi untuk mendeteksi putaran kepingan (*disk*) dari kWh meter induksi yang di tera, dimana pendeteksian yang dilakukan



adalah tanda hitam (*black mark*) pada kepingannya. Output sensor berupa pulsa dimana dalam satu perioda putaran kepingan terdapat dua buah pulsa. Output sensor akan tinggi ("1"), bilamana tanda hitam yang ada di kepingan berada tepat pada input sensor, sedang saat tidak ada tanda hitam pada kepingan, maka output sensor akan rendah ("0"). Pulsa output dari sensor tersebut diumpankan pada mikrokontroler dan digunakan untuk menghitung jumlah pulsa dari kWh meter standar yang masuk mikrokontroler.

KWh meter standar yang dipergunakan disini harus yang mempunyai fasilitas pulsa output, dimana pulsa output tersebut mewakili sekumpulan daya dalam artian pulsa outputnya akan berubah-ubah (bervariasi) terhadap dayanya. KWh meter standar yang digunakan dalam perancangan ini menggunakan merek ENERTEC tipe E4Y21E, dengan besaran pulsa output adalah 10  $\mu\text{Wh.C/pulsa}$ , dan harga konstanta C bervariasi sesuai dengan tegangan dan arus yang digunakan.

## 2. Mode Switch.

Mode switch ini fungsinya untuk memilih menu pengetesan yang dilakukan. Dalam perencanaan dipilih delapan menu, yaitu : Warming Up; 1/20 Ib; 1/10 Ib; 1 Ib; 2 Ib; 4 Ib; 6 Ib; Starting. Warming Up berfungsi untuk pemanasan, karena kWh meter sebelum di tera perlu dilakukan pemanasan lebih dahulu agar putaran kepingan stabil. Sedang pada menu pengetesan 1/20Ib dan 1/10Ib, diperlukan untuk pengetesan pada saat beban rendah (*low load*). Dan pada menu 1 Ib, 2 Ib, 4 Ib, 6 Ib, diperlukan untuk pengetesan

beban penuh ataupun untuk pengetesan kWh meter induksi dengan kapasitas besar. Serta pada menu starting digunakan untuk pengetesan apakah kWh meter tersebut startnya baik atau tidak, ini akan berkaitan dengan saat kWh meter tanpa beban, sebab bila kWh meter tanpa beban, maka kWh meter tersebut harus berhenti atau tidak boleh berputar selama satu putaran.

### **3. Konstanta DIP Switch.**

Pada kWh meter induksi yang akan di tera dan kWh meter standar yang dipergunakan, mempunyai suatu konstanta yang berbeda-beda. Untuk kWh meter standar, konstantanya tergantung pada tegangan dan arus yang diterapkan padanya. Sedangkan pada kWh meter induksi, konstantanya tergantung pada jenis atau mereknya.

Oleh karena itu, agar peralatan penghitung kesalahan kWh meter induksi ini dapat digunakan pada kondisi konstanta yang berbeda-beda tersebut, maka diberikan suatu fasilitas DIP Switch yang dapat diseting sesuai dengan konstanta-konstanta yang ada, dalam hal ini konstanta-konstanta pada kWh meter induksi dan atau Kwh meter standar. Seting pada DIP Switch tersebut merupakan besaran acuan yang akan digunakan oleh mikrokontroler untuk dibandingkan dengan jumlah pulsa dari kWh meter standar.

### **4. Mikrokontroler.**

Mikrokontroler disini merupakan jantung dari alat penghitung kesalahan kWh meter induksi yang direncanakan, semua output-output dari sensor,

kWh meter standar, mode switch, serta konstanta DIP switch, semua itu diumpankan ke mikrokontroler. Dari mikrokontroler ini, semua keadaan ataupun kondisi input-input akan dideteksi sesuai dengan urutan program yang telah dimasukkan dalam mikrokontroler tersebut. Semua input-input tersebut merupakan data yang akan disimpan didalam memori yang tersedia. Kemudian data-data tersebut diolah sesuai dengan keadaan mode switch (menu pengetesan) dan akan ditampilkan pada led-led indikator ataupun *sevent segment*.

#### **5. Tampilan.**

Untuk menampilkan hasil dari proses alat penghitung kesalahan kWh meter induksi ini terdiri dari tiga macam, yaitu :

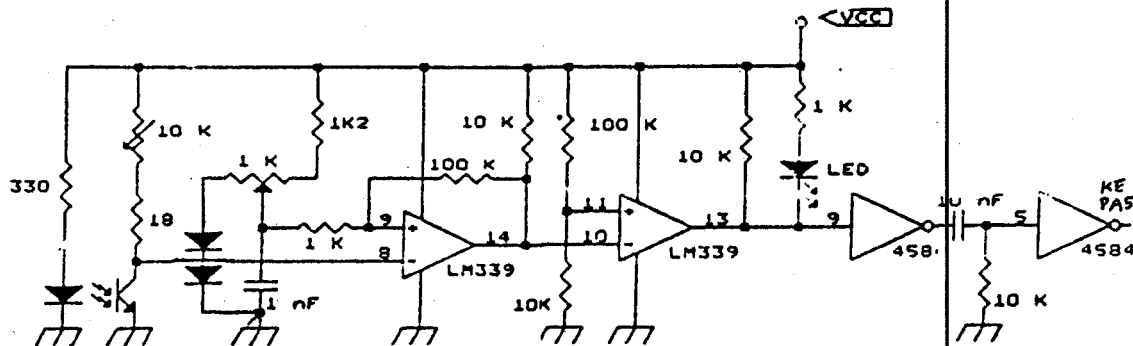
- a. Untuk keadaan menu pengetesan, ditampilkan berupa penyalan dari led-led indikator.
- b. Untuk prosentase kesalahan, ditampilkan oleh 7 segment dalam dua digit.
- c. Untuk tingkat kesalahan positif ataupun negatif, ditampilkan dengan penyalan salah satu dari dua buah led indikator yang terletak pada bagian kiri atas dari peralatan ini.

#### **IV.2. Perangkat Keras**

Pada pembuatan perangkat keras untuk penghitung kesalahan KWh meter induksi, sesuai dengan perencanaan blok diagram diatas, yaitu dengan mengelompokkan pembuatan alat pada masing-masing bloknnya.

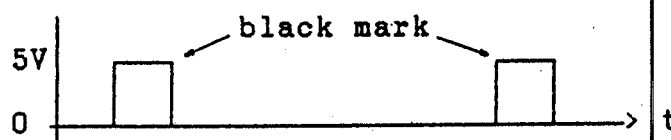
#### IV.2.1. Sensor

Rangkaian penghitung kesalahan kWh meter dilengkapi dengan rangkaian sensor berikut pengkondisi sinyalnya, seperti terlihat pada gambar 4.3.



GAMBAR 4.3  
RANGKAIAN SENSOR

Sensor menggunakan photo transistor dan led (*light emitting diode*), kemudian dikuatkan dengan menggunakan IC LM 339 (*opamp*) dan selanjutnya output dari penguat sensor dikondisikan oleh IC MC 4584 (*Schmitt trigger not gate*), agar keadaan sinyalnya benar-benar pada kondisi yang diinginkan yang sesuai dengan input mikrokontroler. Output tersebut dapat dilihat aktif atau tidaknya melalui led indikator, bila tidak ada tanda hitam pada kepingan, maka led akan menyala, sedang bila ada tanda hitam led akan padam, sedang bentuk pulasanya dapat dilihat pada gambar 4.4.

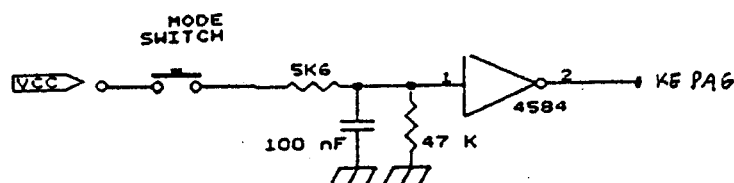


GAMBAR 4.4  
BENTUK OUTPUT SENSOR

Dari perancangan yang diinginkan, bila photo transistor menerima sinar, maka output pengkondisi sinyal pada posisi rendah ("0"); setelah photo transistor tidak ada sinar, kondisi output menjadi tinggi ("1"). Perubahan ada tidaknya sinar, pada rangkaian ini tergantung dari tanda hitam pada kepingan (*black mark*) kWh meter yang sedang di tera. Output tersebut langsung dihubungkan pada MC 68705 P3 melalui PA5, dari sini mikrokontroler akan mendeteksi putaran dari kepingan (dari tanda hitam kepingan).

#### IV.2.2. Mode Switch

Pada rangkaian ini hanya terdapat satu push bottom switch, dan rangkaian *debouncing* yang dibangun dari IC MC4584 (*schmitt trigger not gate*). Rangkaian *debouncing* diperlukan untuk menghindari bentuk pulsa yang tidak diinginkan apabila mode switch ditekan, jadi tidak akan mengganggu kerja, dan outputnya langsung dihubungkan pada mikrokontroler melalui pin 26 (PA 6), serta akan diterjemahkan untuk memilih menu, rangkaian tersebut dapat dilihat pada gambar 4.5.



GAMBAR 4.5  
RANGKAIAN MODE SWITCH

Mode switch ini fungsinya untuk memilih menu pengetesan yang dilakukan. Dalam perencanaan dipilih delapan menu, yaitu : Warming up; 1/20Ib; 1/10Ib; Ib; 2Ib; 4Ib; 6Ib; Starting.

Menu-menu yang disediakan, disesuaikan dengan kondisi pengetesan yang sering dilakukan dalam peneraan kWh meter induksi.

Warming up berfungsi untuk pemanasan, karena kWh meter sebelum di tera perlu dilakukan pemanasan lebih dahulu agar pertaran kepingan stabil. Sedang pada menu pengetesan 1/20Ib dan 1/10Ib, diperlukan untuk pengetesan pada saat beban rendah (*low load*). Dan pada menu Ib, 2Ib, 4Ib, 6Ib, diperlukan untuk pengetesan beban penuh ataupun untuk pengetesan kWh meter induksi dengan kapasitas besar. Serta pada menu starting digunakan untuk pengetesan apakah kWh meter tersebut startnya baik atau tidak, ini akan berkaitan dengan saat kWh meter tanpa beban, sebab bila kWh meter tanpa beban, maka kWh meter tersebut harus berhenti atau tidak boleh berputar selama satu putaran.

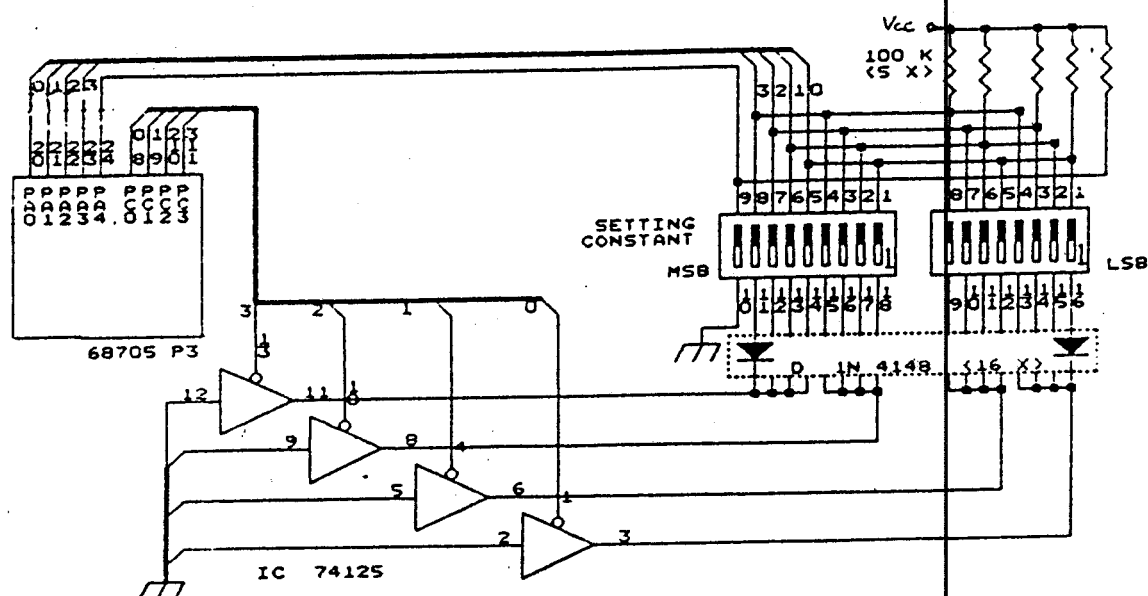
#### IV.2.3 Konstanta DIP Switch

Rangkaian ini terdiri dari sederetan DIP switch yang jumlahnya ada 17 buah, dalam perencanaannya dimaksudkan dengan jumlah tersebut sudah dapat mewakili konstanta-konstanta yang ada pada kWh meter induksi dan pada kWh meter standar. Semua DIP switch tersebut dikendalikan oleh mikrokontroler melalui IC 74125 (*buffer tristate*).

Karena fasilitas yang tersedia pada port C dari mikrokontroler

jumlahnya hanya 4-bit, sedangkan jumlah bit konstanta ada 17-bit, maka proses pembacaannya tidak dapat secara langsung melainkan dengan cara dibaca secara bergantian masing-masing 4-bit. Pengontrolan pembacaan tersebut dilakukan oleh port C melalui buffer tristate, sedang 1-bit terakhir langsung dibaca, tanpa melalui pengontrolan seperti diatas.

Semua output dari DIP switch langsung dihubungkan ke mikrokontroler melalui port A (PA0-PA4), dan langsung disimpan pada register dan dianggap sebagai besaran acuan untuk dibandingkan dengan jumlah pulsa yang masuk mikrokontroler dari kWh meter standar. Rangkaian tersebut dapat dilihat pada gambar 4.6 berikut.



**GAMBAR 4.6**  
**RANGKAIAN KONSTANTA DIP SWITCH**

#### IV.2.4 Mikrokontroler

Mikrokontroler disini merupakan jantung dari alat penghitung kesalahan kWh meter induksi yang direncanakan, semua output-output dari sensor, kWh meter standar, mode switch, serta konstanta DIP switch, semua itu diumpankan ke mikrokontroler. Dari mikrokontroler ini, semua keadaan ataupun kondisi input-input akan dideteksi sesuai dengan urutan program yang telah dimasukkan dalam mikrokontroler tersebut. Semua input-input tersebut merupakan data yang akan disimpan didalam memori yang tersedia. Kemudian data-data tersebut diolah sesuai dengan keadaan mode switch (menu pengetesan) dan akan ditampilkan pada led-led indikator ataupun *sevent segment*.

Mikrokontroler yang dipakai adalah IC MC 68705 P3, karena sesuai dengan yang dibutuhkan, baik pada perangkat keras maupun pada perangkat lunak, rangkaian tersebut dapat dilihat pada gambar 4.7 berikut.

Pada rangkaian ini perancangannya dibuat sedemikian rupa sehingga mampu mencakup fungsi kerja yang diinginkan, yang disesuaikan pula dengan program yang dimasukkan didalamnya.

IC MC 68705 P3 dioperasikan dengan menggunakan kristal 4 MHz. yang banyak dijumpai dipasaran. Port A (PA0 - PA6) dioperasikan sebagai input, sedang PA7 dioperasikan sebagai output. Input-input disini (PA0 - PA4) berfungsi untuk inisialisasi konstanta kWh meter melalui pensetingan DIP switch dengan pengontrolan IC 74125 (*buffer tristate*). Sedang input PA5, sebagai input dari pulsa sensor, dan PA6 sebagai input dari mode



switch untuk memilih konfigurasi menu pengetesan kWh meter. Sedang PA6 sebagai output untuk menunjukkan tingkat kesalahan kWh meter positif atau negatif. Port PB0 - PB2 difungsikan sebagai output, untuk indikator menu pengetesan kWh meter, sedangkan PB3 - PB6 difungsikan sebagai output untuk menunjukkan besarnya persen kesalahan kWh meter melalui display seven segment yang sebelumnya di decoder lebih dahulu dengan IC 4511 (*BCD to seven segment*).

Saat catu daya di-ON-kan, yang dalam hal ini MC 68705 P3 mendapat catu tegangan  $V_{cc} +5$  volt, yang dapat dianggap sebagai pemicu, maka dalam hal ini tegangan ambang (pin6) dan pemicu (pin7) masing-masing masih dibawah VUT dan VLT, sehingga keluaran menjadi tinggi. Dengan mulai diisinya muatan kapasitor, tegangan pemicu akan naik, sedangkan tegangan keluaran tetap tinggi meskipun pemicu telah melewati VLT. Hal ini disebabkan karena tegangan ambang masih dibawah VUT. Beberapa selang waktu kemudian, baik tegangan pemicu maupun tegangan ambang akan berada diatas VUT sehingga menyebabkan tegangan keluaran menjadi rendah. Selang waktu ( $t$ ) antara saat pengisian kapasitor sampai tegangan pemicu dan ambang mencapai diatas VUT inilah yang merupakan waktu tundanya.

Pada MC 68705 P3, kondisi power-ON-reset menimbulkan waktu tunda tertentu. Dalam artian, pada pin RESET akan tetap rendah dengan selang waktu  $t_{RHL}$  sebelum kondisi berubah menjadi tinggi. Dengan memberi kapasitor  $1 \mu F$  pada pin reset akan didapat  $t_{RHL} = 100$  msec.

Sedang pulsa sinyal dari standar meter diumpankan pada pin 2 dari

mikrokontroler (INT), sinyal ini dianggap sebagai sinyal penyelaan kemudian disimpan pada index register (X), sesuai dengan program.

Pada pin 25 (PA5) mendapat masukan dari keluaran sensor, ini akan diterjemahkan sebagai sinyal start-stop, berapa jumlah pulsa dari standar meter yang masuk.

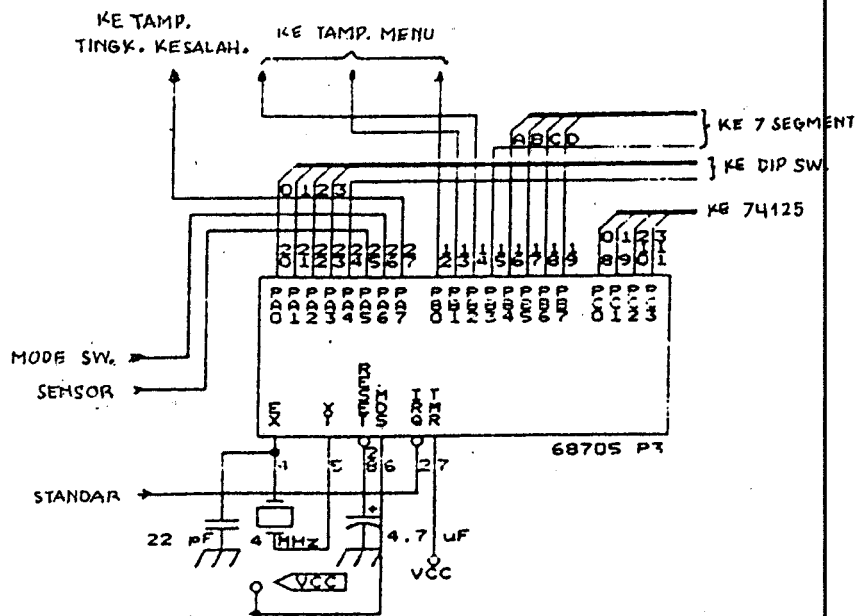
Sedang pada pin 26 (PA6) mendapat masukan dari sinyal mode switch, yang akan ditanggapi oleh program untuk memindahkan kondisi pengetesan.

Saat MC 68705 P3 sudah mendapat catu daya, beberapa saat kemudian keluaran port C (PC0 - PC3) akan aktif secara bergantian, untuk mendeteksi posisi dari DIP switch melalui IC 74125 (*buffer tristate*) dan posisi tersebut akan dibaca oleh PA0 - PA4. Karena jumlah DIP switch ada 17 bit, sedang port hanya mampu menangani 5 bit, maka proses pembacaannya adalah : bit 17 langsung dibaca oleh PA4, sedang yang 16 bit berikutnya, dibaca dengan cara bergeser masing-masing 4 bit. Kemudian setelah semua data pada DIP switch terbaca semua, data tersebut akan disimpan pada *index register (X)*.

Setelah semua masukan ditanggapi dan berada dalam memori, maka akan dengan segera diproses sesuai dengan menu yang telah diprogramkan dan kemudian hasilnya akan ditampilkan pada tampilan seven segment melalui IC MC 4584 (*decoder BCD to seven segment*), berikut tingkat prosentase dari perhitungan, baik positif atau negatif.

Semua urutan kerja atau proses diatas, baik input-input penanganan

pulsa standar meter, sinyal dari sensor, sinyal dari mode switch, data dari DIP switch; dan output-output prosen negatif maupun positif, juga tampilan menu, semua itu dikendalikan oleh program.

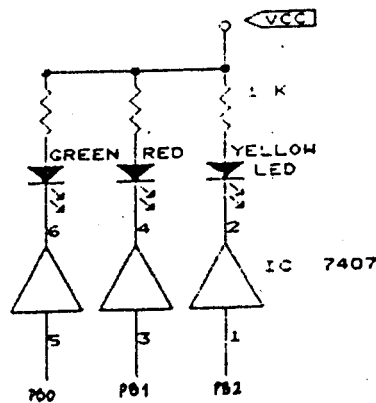


GAMBAR 4.7  
RANGKAIAN MIKROKONTROLER

#### IV.2.5. Tampilan

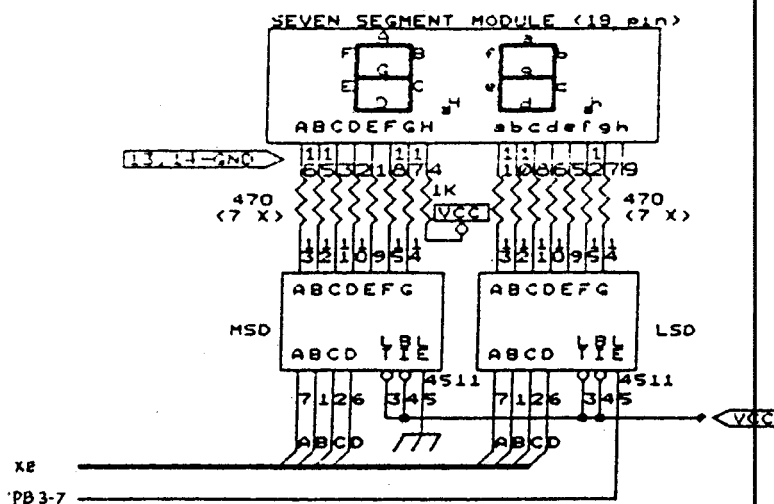
Tampilan dari semua hasil proses yang telah dilakukan oleh mikro-kontroler, terdiri atas tiga bagian, yaitu :

- Keadaan menu pengetesan, ditampilkan berupa penyalan 3 buah led indikator, dimana penyalan led-led tersebut dikendalikan oleh port B (PB0-PB2) melalui IC 7407 (buffer), dan rangkaiannya seperti pada gambar 4.8.



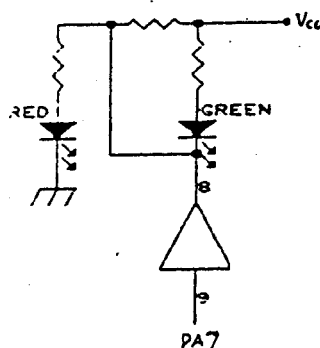
GAMBAR 4.8  
RANGKAIAN TAMPILAN MENU PENGETESAN

- b. Prosentase kesalahan, ditampilkan oleh dua buah 7 segment, karena port B (PB4-PB7) yang merupakan output untuk 7 segment hanya 4-bit sedangkan prosentasenya dua digit dan masih dalam bentuk BCD (*binary code decimal*), maka output tersebut diubah terlebih dahulu melalui dua buah IC MC 4511 (*BCD to seven segment*).



GAMBAR 4.9  
RANGKAIAN TAMPILAN PROSENTASE KESALAHAN

- c. Tingkat kesalahan positif ataupun negatif, ditampilkan dengan penyalan salah satu dari dua buah led indikator yang diambil dari output PA7 melalui IC 7407 dan rangkaiannya terlihat pada gambar 4.10.



GAMBAR 4.10  
RANGKAIAN TAMPILAN TINGKAT KESALAHAN

Dari semua rangkaian-rangkaian diatas, terdapat juga rangkaian regulator tegangan dengan menggunakan IC LM 7805 dengan besar tegangan outputnya +5 volt. Tegangan ini untuk catu seluruh rangkaian.

Keseluruhan rangkaian dari alat penghitung kesalahan kWh meter ini direalisasikan pada dua buah papan rangkaian tercetak dengan tata letak komponen yang kompak. Gabungan rangkaian tersebut terdapat pada gambar 4.11 dan 4.12.





### IV.3. Perangkat Lunak

Sesuai dengan pembatasan masalah pada bab pendahuluan, perencanaan perangkat lunak meliputi pembuatan program penghitung kesalahan kWh meter dengan menggunakan bahasa assembly, yang penulisannya memakai program penulisan *Side kick*. Kemudian di *assembler* kan dengan memakai program XASM05, dan selanjutnya diubah dengan menggunakan program HO (*high low*) untuk dapat diisikan dalam EPROM. Setelah semua program masuk dalam EPROM, kemudian dengan perangkat pemrograman MC 68705 P3 semua datanya ditransfer dalam IC MC 68705 P3. Pada subbab ini akan menerangkan diagram alir secara garis besar.

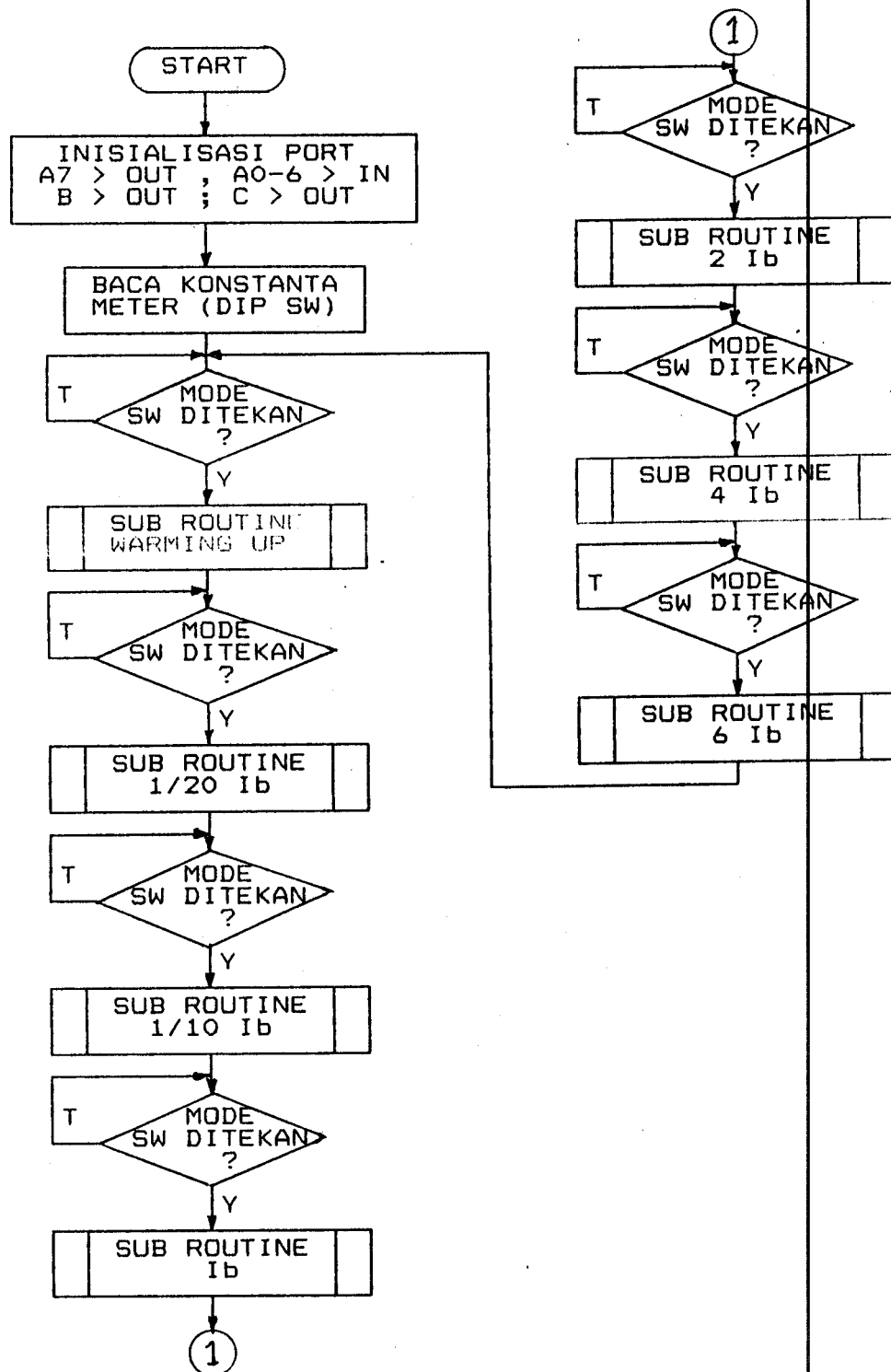
#### IV.3.1. Diagram Alir

Diagram alir penghitung kesalahan kWh meter secara garis besar seperti yang ditunjukkan pada gambar 4.14. Proses tersebut diawali dengan pembacaan program yang dimulai dari alamat  $80_H$ .

Kemudian setelah start mikrokontroler melakukan inisialisasi port, dimana port A7 sebagai output, sedang port A6 - A0 sebagai input, dan port B sebagai output, juga port C sebagai output. Setelah masing-masing port siap difungsikan, selanjutnya melakukan proses seperti terlihat pada diagram alir berikut.



## PROGRAM UTAMA



GAMBAR 4.13  
DIAGRAM ALIR PENGHITUNG KESALAHAN KWH METER

Sedang routine program untuk inisialisasi port sebagai berikut :

```

      ORG 80H      ; Start address at 80H
START LDA #$FF
      STA PORTC
      STA DDRB    ; PORT B as OUTPUT
      STA DDRC    ; PORT C as OUTPUT
      CLR PORTA
      LDA #80H
      STA DDRA    ; PORT A7 as OUTPUT; A6-0 as INPUT
      ORA #07H
      STA PORTB
  
```

Selanjutnya membaca konstanta kWh meter, dengan routine program-nya adalah:

```

      CLR TEMP
      CLR CONST2
      BCLR 0,PORTC
      LDA #98      ; DELAY DAHULU 0.2 DETIK
      STA TUNDA
      JSR DELAY1
LBL1  LDA PORTA    ; START BACA KONSTANTA METER
      AND #0FH    ; CATATAN
      BRSET 0,TEMP,LBL2
      STA TEMP0    ; KONSTANTA METER ADALAH 17 BIT
      BRA LBL3    ; MERUPAKAN JMLH PULSA PD 1/101b
LBL2  LSLA        ; BIT 16 PADA CONST2
      LSLA        ; BIT 15 S/D 8 PADA CONST1
      LSLA        ; BIT 7 S/D 0 PADA CONST0
      LSLA
      ORA TEMP0
      LDX TEMP
      LSRX
      STA CONST1,X
LBL3  INC TEMP
      SEC
      ROL PORTC
      LSR PORTB
      BCLR 3,PORTB
      BSET 3,PORTB
      LDA #98      ; DELAY DAHULU 0.2 DETIK
      STA TUNDA
  
```

```

JSR DELAY1
LDX #04H
CPX TEMP
BNE LBL1
LDA CONST1
LDX CONST0
STA CONST0
STX CONST1
BRCLR 4,PORTA,LBL4
BSET 0,CONST2
BRA MENU0
LBL4  BCLR 0,CONST2      ; FINISH BACA KONSTANTA METER

```

Setelah proses pembacaan konstanta meter selesai, kemudian memilih mode yang diinginkan dengan routine programnya adalah sebagai berikut :

```

MENU0  RSP
        BSET 7,PORTA
MENU01 BRSET 6,PORTA,MENU01
        BSR WARMUP      ; RUNNING SUB-ROUTINE WARMING-UP
MENU02 BRCLR 6,PORTA,MENU02
MENU03 BRSET 6,PORTA,MENU03
MENU1  BRCLR 6,PORTA,MENU1
        BSR S1IB20      ; RUNNING SUB-ROUTINE 1/20 Ib
MENU2  BRCLR 6,PORTA,MENU2
        JSR S1IB10      ; RUNNING SUB-ROUTINE 1/10 Ib
MENU3  BRCLR 6,PORTA,MENU3
        JSR S1IB        ; RUNNING SUB-ROUTINE 1b
MENU4  BRCLR 6,PORTA,MENU4
        JSR S2IB        ; RUNNING SUB-ROUTINE 2 * Ib
MENU5  BRCLR 6,PORTA,MENU5
        JSR S4IB        ; RUNNING SUB-ROUTINE 4 * Ib
MENU6  BRCLR 6,PORTA,MENU6
        JSR S6IB        ; RUNNING SUB-ROUTINE 6 * Ib
MENU7  BRCLR 6,PORTA,MENU7
        JSR STARTING    ; RUNNING SUB-ROUTINE STARTING
MENU8  BRSET 6,PORTA,MENU8
        LDA #07H        ; DISPLAY-KAN 00
        STA PORTB       ; SEMUA LED MENU OFF
MENU9  BRCLR 6,PORTA,MENU9
        BRA MENU0       ; AKHIR PROGRAM UTAMA

```

Langkah-langkah kerja dari mikrokontroler selanjutnya yang lebih lengkap sesuai dengan routine program, maupun sub routine program berikutnya dapat dilihat pada lampiran dalam bentuk diagram alir.

#### IV.4. Cara Kerja Alat Penghitung Kesalahan KWh Meter

Sebelum mengoperasikan alat penghitung kesalahan kWh meter ini, disini perlu diketahui lebih dahulu dasar perhitungan kesalahan dari alat tersebut, yaitu dengan memperhatikan konstanta dari kWh meter standar yang digunakan dan konstanta kWh meter induksi yang sedang di tera.

Dalam perhitungan yang dilakukan disini, menggunakan kWh meter standar dengan tipe E4Y21E, buatan ENERTEC, Schlumberger, Perancis, yang mempunyai besaran pulsa output 10  $\mu$ Wh C/pulsa.

Dengan pemisalan konstanta kWh meter standar (C), yang digunakan adalah pada saat  $I = 5$  A dan  $V = 220$  V, yaitu  $C = 11$ , dan konstanta kWh yang akan di tera buatan Mecoindo dengan tipe M2X dan kemampuannya 5 (20) A, 220 V, dengan  $k = 1250$  putaran/kWh, maka :

$$\begin{aligned} \text{jumlah pulsa} &= \frac{1}{k C} \\ &= \frac{1}{1250 \text{ kWh/putaran} \cdot 11 \cdot 10^{-5} \text{ Wh/pulsa}} \\ &= \frac{10^8}{1250 \cdot 11} \left[ \frac{\text{pulsa}}{\text{putaran}} \right] \\ &= 72727 \left[ \frac{\text{pulsa}}{\text{putaran}} \right] \end{aligned}$$

Bila diubah dalam bentuk bilangan hexadesimal, maka:

$$\text{jumlah pulsa} = 11C17_{(H)} \left[ \frac{\text{pulsa}}{\text{putaran}} \right]$$

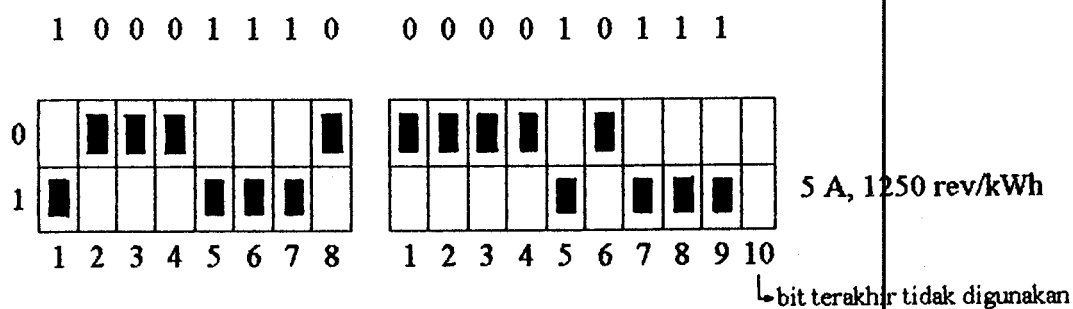
Setelah mengetahui jumlah pulsa/putaran, maka perlu diketahui juga cara mensetting DIP switch, yaitu sebagai berikut :

Jumlah pulsa/putaran dalam bentuk hexadesimal terlebih dahulu diubah menjadi bilangan biner, yaitu :

1 1 C 1 7 → hexadesimal

0001 0001 1100 0001 0111 → biner

Jadi posisi DIP switchnya adalah seperti gambar 4.14.



GAMBAR 4.14  
POSISI SETTING DIP SWITCH

Setelah menghitung konstanta, maka selanjutnya memasukkan atau mensettingkan DIP switch.

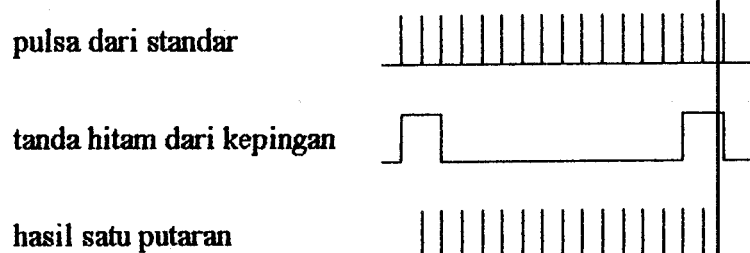
Sedang perhitungan prosen kesalahan dihitung berdasarkan sampling pulsa yang masuk, dan hasil perhitungan yang dimasukkan/disetting melalui DIP switch, sebagai contoh misalnya pulsa hasil perhitungan = X, sedang pulsa hasil sampling = Y, maka beda kesalahan = Y - X

bila beda kesalahan = +    maka kesalahan - (meter pelan)

bila beda kesalahan = -    maka kesalahan + (meter cepat)

prosen kesalahan =  $\frac{\text{beda kesalahan}}{Y} \times 100 \%$

Dibawah ini, digambarkan contoh sampling pulsa :



GAMBAR 4.15  
SAMPLING PULSA

Selanjutnya pertama-tama saat rangkaian dinyalakan, maka kondisi yang tampak adalah display akan menyala nol, dan indikator led minus akan menyala. Lalu dengan menekan saklar mode, untuk memilih konfigurasi menu dengan memperhatikan indikator led menu, seperti terlihat pada tabel berikut.

TABEL 4.1  
KONFIGURASI MENU

MODE PENGETESAN	INDIKATOR LED		
Warming Up	mati	mati	mati
1/20 Ib (5% Ib)	mati	mati	nyala
1/10 Ib (10% Ib)	mati	nyala	mati
1 Ib	mati	nyala	nyala
2 Ib	nyala	mati	mati
4 Ib	nyala	mati	nyala
6 Ib	nyala	nyala	mati
Starting	nyala	nyala	nyala

Dari menu tersebut kita dapat memilih mode pengetesan, bila kWh meter saat pertama kali di tes, maka perlu ada pemanasan selama 45 menit. Pada kondisi ini alat penghitung berfungsi sebagai jam, yaitu untuk mengetahui lamanya pemanasan. Pada kondisi ini putaran dari kepingan tidak dihiraukan dalam artian tidak dideteksi.

Bila kWh meter sudah dipanasi, maka langkah selanjutnya memilih pada kondisi pengetesan yang diinginkan, dengan menekan saklar mode.

Pada pengukuran kesalahan kWh meter, pada kondisi 1/20Ib, 1/10Ib, 1Ib, alat akan mendeteksi kepingan selama satu putaran, dan pada tampilan akan menunjukkan besar kesalahan dalam prosen setiap satu putaran. Sedang pada kondisi 2 Ib, alat akan mendeteksi setiap dua putaran kepingan dan pada kondisi 4 Ib akan mendeteksi setiap empat putaran kepingan, serta pada kondisi 6 Ib akan mendeteksi setiap enam putaran kepingan, kemudian prosen kesalahan akan ditampilkan.

Sedang pada kondisi starting, alat akan mendeteksi satu putaran kepingan, dan alat akan berfungsi sebagai jam dimana pada kondisi starting ini kepingan akan berputar dan bila starting dari meter baik, maka tampilan akan menunjuk angka dibawah 15, kemudian berkedip (kedipnya lambat). Tetapi bila meter kurang baik, maka tampilan akan menunjuk angka 15 dan kemudian berkedip (kedipnya cepat).

## BAB V

### KESIMPULAN

Dari seluruh pembahasan dari Tugas Akhir ini, akan diambil suatu kesimpulan sebagai berikut :

1. Penggunaan alat penghitung kesalahan kWh meter, dapat meningkatkan efektifitas kerja. Karena selain menampilkan besar kesalahannya dengan segera atau langsung terbaca, kita dapat juga melakukan kompensasi-kompensasi pada kWh meter induksi yang sedang di tera, tanpa harus mematikan kWh meter standar.
2. Dengan menu pengetesan yang tersedia, kita dapat memilih kondisi pengetesan sesuai dengan yang diinginkan.
3. IC MC 68705 P3 dibandingkan dengan minimum sistem akan lebih mudah menerapkan, dan dari segi ekonomis lebih unggul.
4. Dengan adanya EPROM didalamnya, IC MC 68705 P3 mempunyai kelebihan bahwa bila sudah diprogram, maka isi dari program tersebut tidak dapat dilihat, dengan demikian kerahasiaan program akan lebih terjamin.
5. Kelebihan lain dari IC MC 68705 P3 ini, dapat diprogram dengan menggunakan peralatan yang sederhana, sehingga lebih mudah mengaplikasikan dibanding dengan jenis mikrokontroler lainnya.



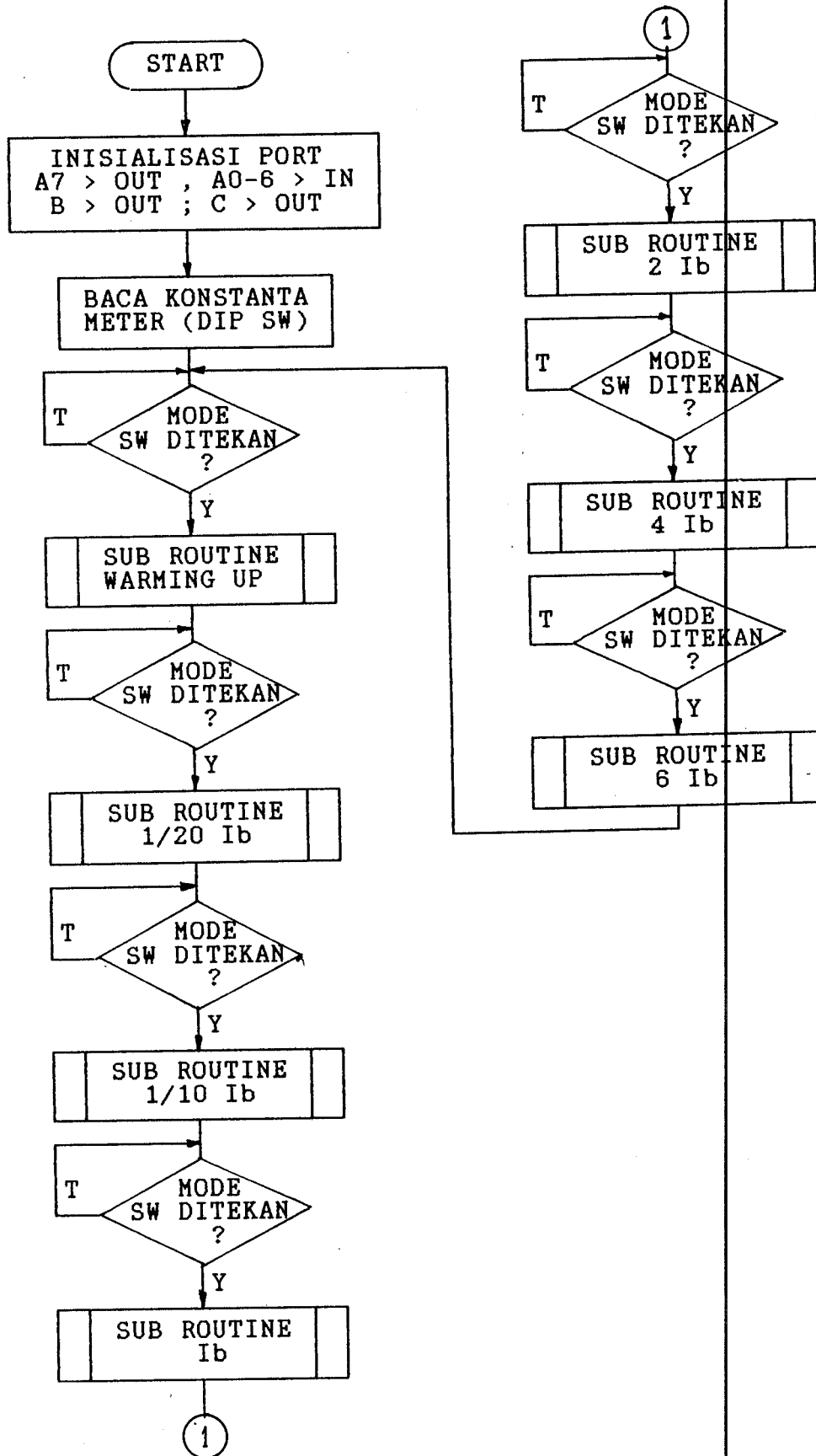
## DAFTAR PUSTAKA

1. BL. Theraja, **"Electrical Technology"**, Nirja Construction & Development, Co., (P) Ltd., New Delhi, 1984.
2. Busono, **"Mikrokontroler MC68705U3: Panduan Pembuatan Program dan Rangkaian"**, Penerbit Dinastindo, Jakarta, 1992.
3. Frank A Laws, SB., **"Electrical Measurements"**, McGraw Hill Book Company, Inc., New York & London, 1917.
4. Fuller, G, **"Mark Space Amplitude Multiplier"**, Schlumberger, Perancis, 1987.
5. Greenfield, Joseph D.; William C. Wray, **"Using Microprocessor and Microcomputers: The Motorola Family"**, Second Edition, John Wiley & Son, Inc., Canada 1988.
6. Isaac F Kinnard, DSc., **"Applied Electrical Measurement"** John Wiley & Son, Inc., New York., Chapman & Hall. Ltd., London, 1956.
7. Peatman, John B., **"Design with Microcontrollers"**, McGraw-Hill Book Company, Singapore, 1988.
8. Soedjana Sapi'i, DR., Osamu Nishino, DR., **"Pengukuran dan Alat-alat Ukur Listrik"**, Pradnya Paramita, Cetakan Ketiga, Jakarta, 1979.
9. Steeman, J. P. M., **"Data Sheet Book"**, PT. Media Komputindo Gramedia, Jakarta, 1988.
10. Zaks, Rodnay;(pen; S.H. Nasution), **"Dari Chip ke Sistem: Pengantar Mikroprosesor"**, Penerbit Erlangga, Jakarta, 1986.

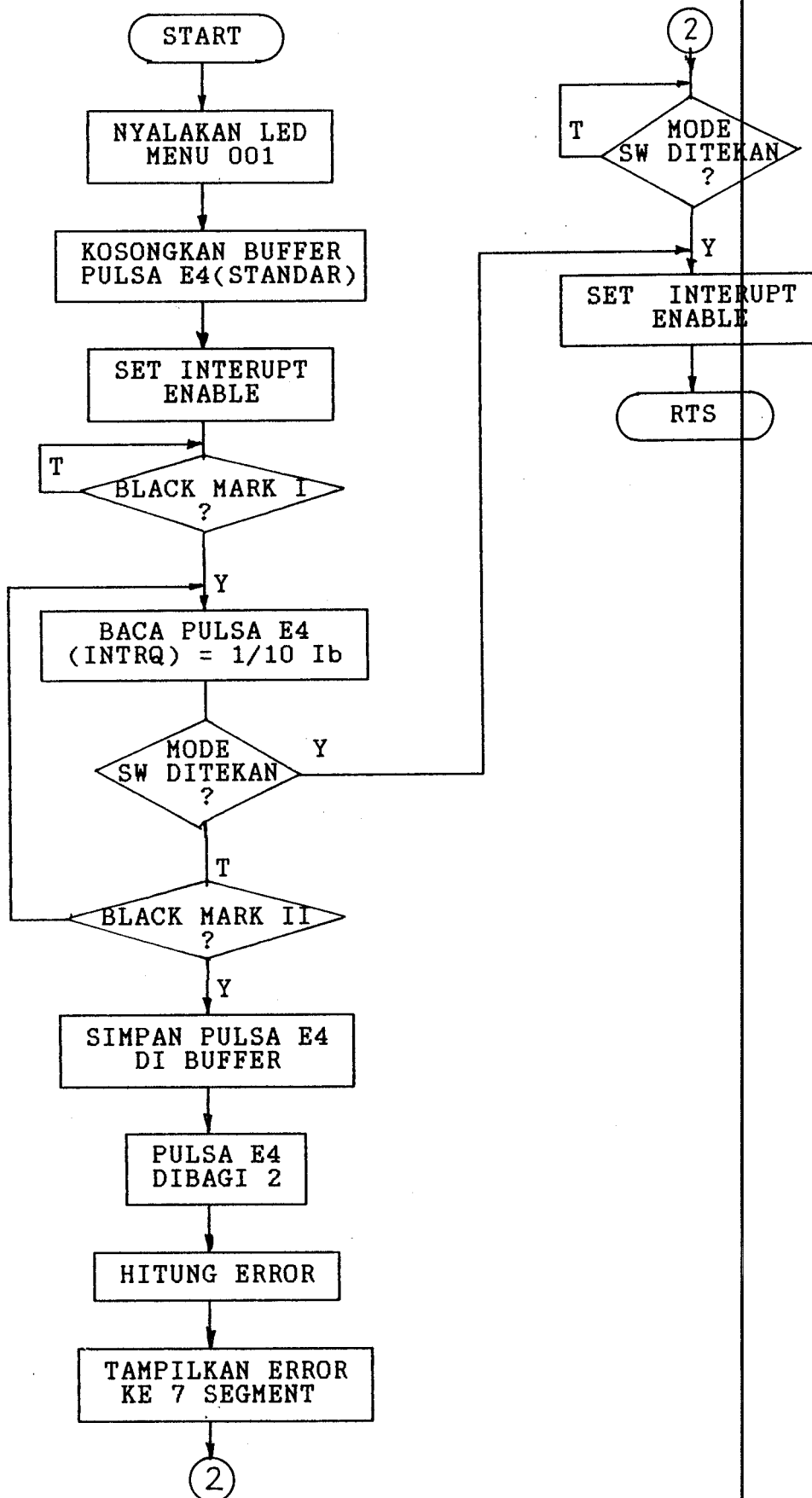
11. -----, **"Peraturan Peneraan kWh Meter Arus bolak-balik"**, PLN., LMK., Jakarta.
12. -----, **"IEC Standard"**, Publication 521, First edition, IEC Standard, Jenewa, Swiss, 1976.
13. -----, **"HC05 M68HC05 Applications Guide"**, Motorola Inc., USA., 1991.
14. -----, **"MC68705P3/R3/U3 8-BIT Microcomputer Programming Module"**, Application Note, AN.857 Rev2., Motorola Inc., USA., 1985.
15. -----, **"Motorola Microprocessor Data Book"**, Motorola Inc., USA., 1989.

## LAMPIRAN

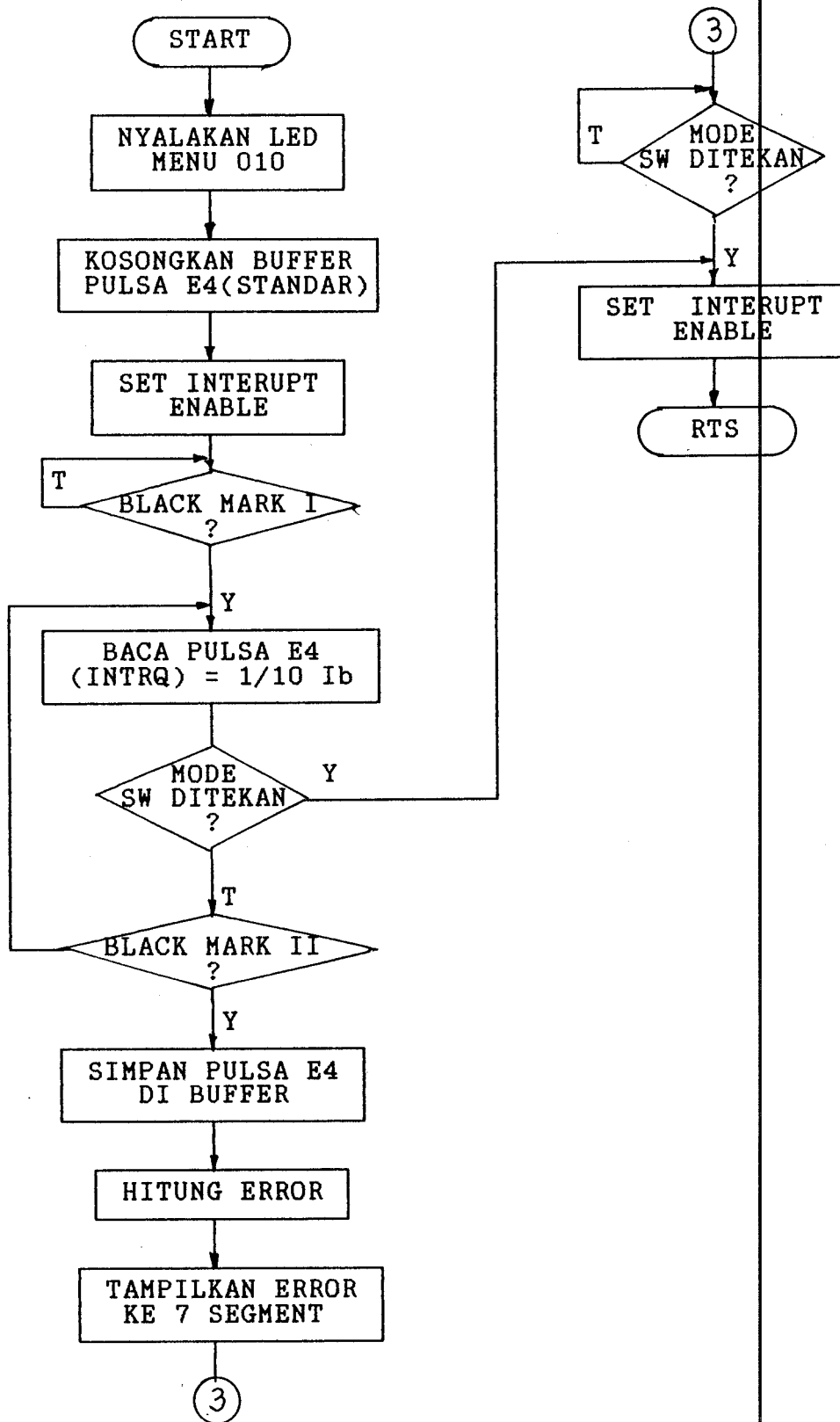
PROGRAM UTAMA



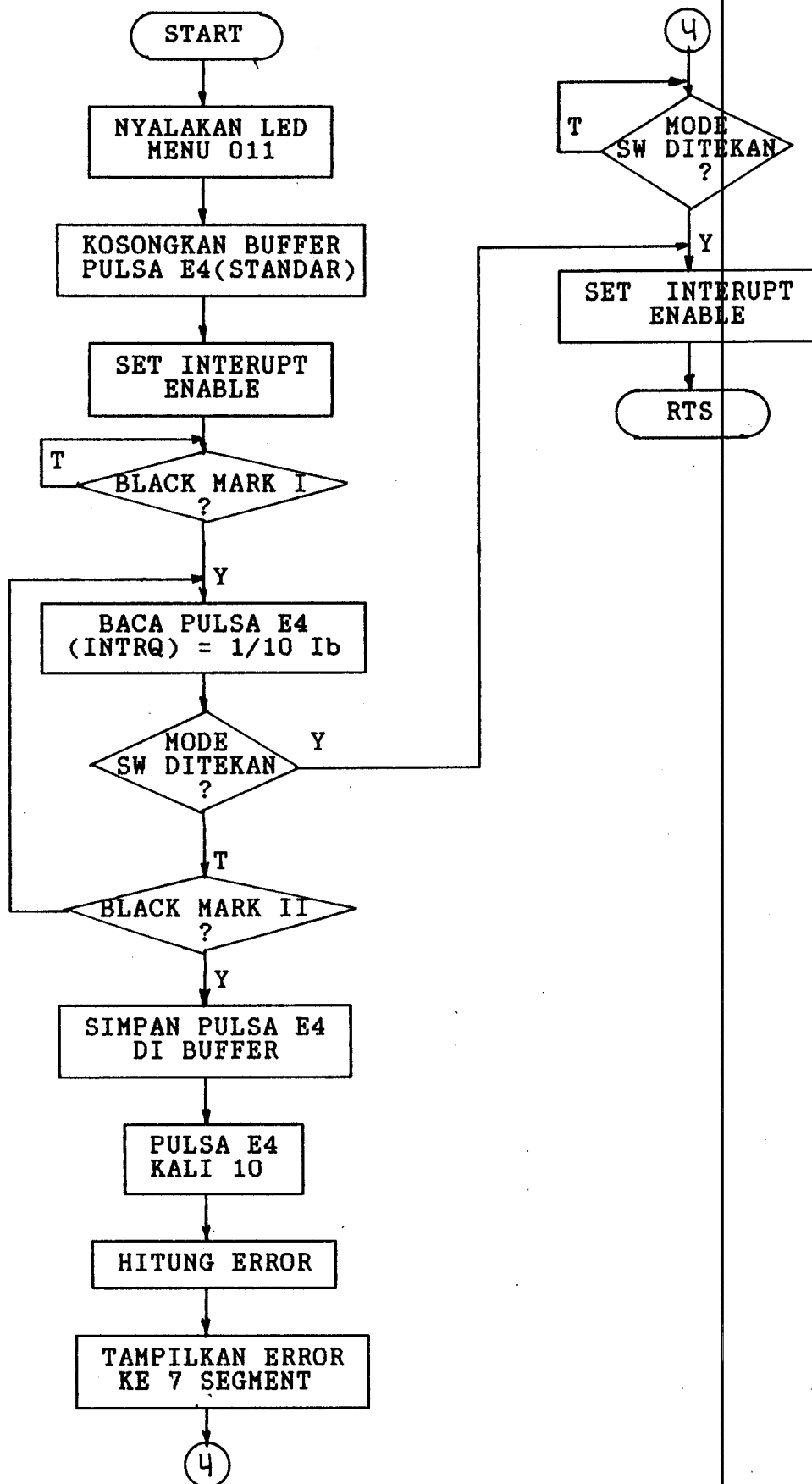
SUB ROUTINE 1/20 Ib



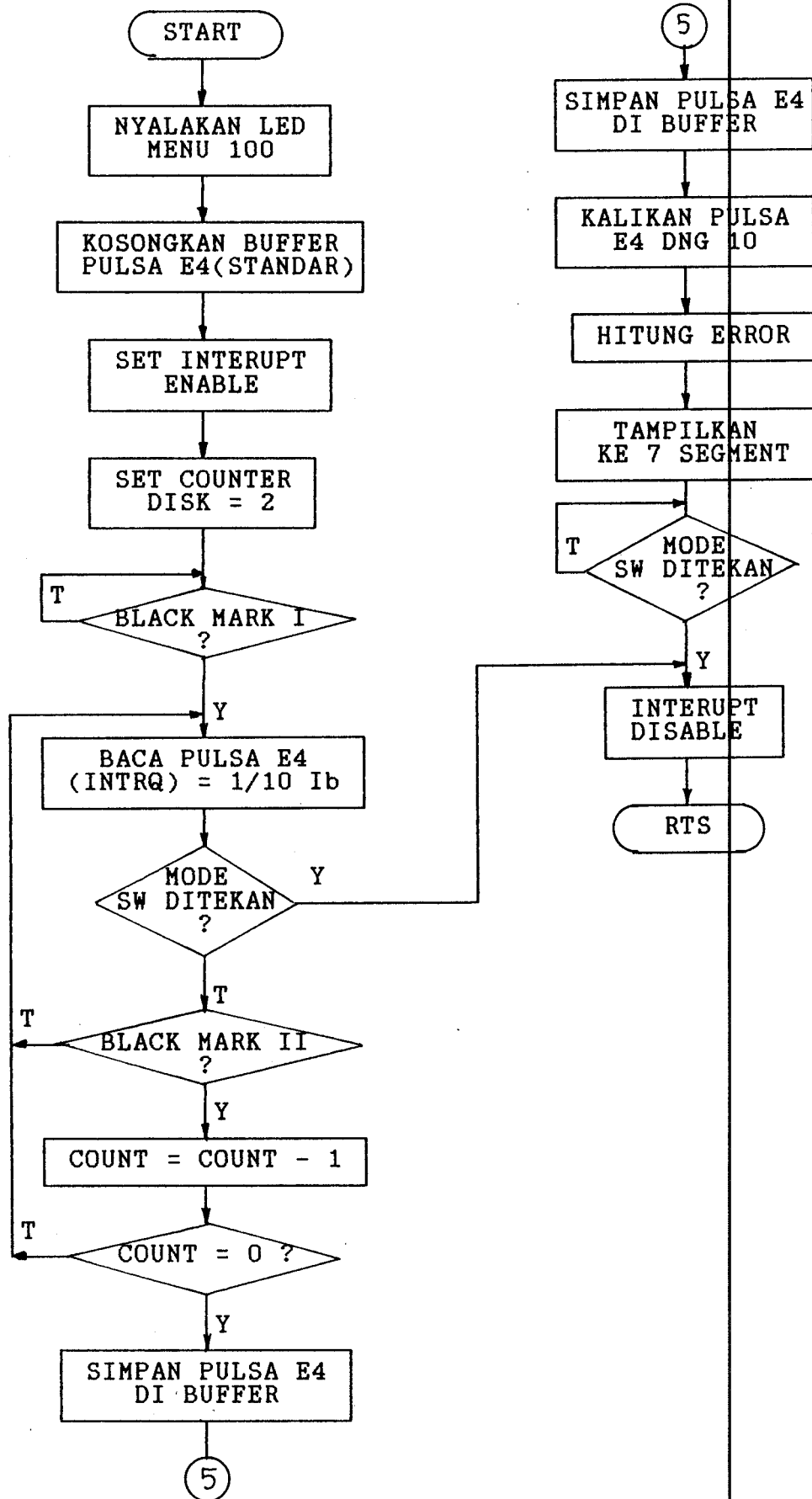
SUB ROUTINE 1/10 Ib



## SUB ROUTINE Ib

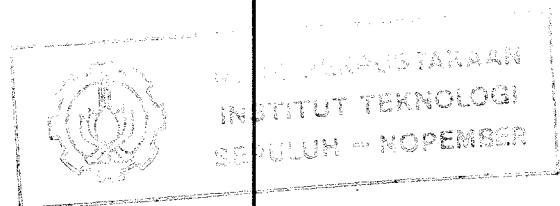
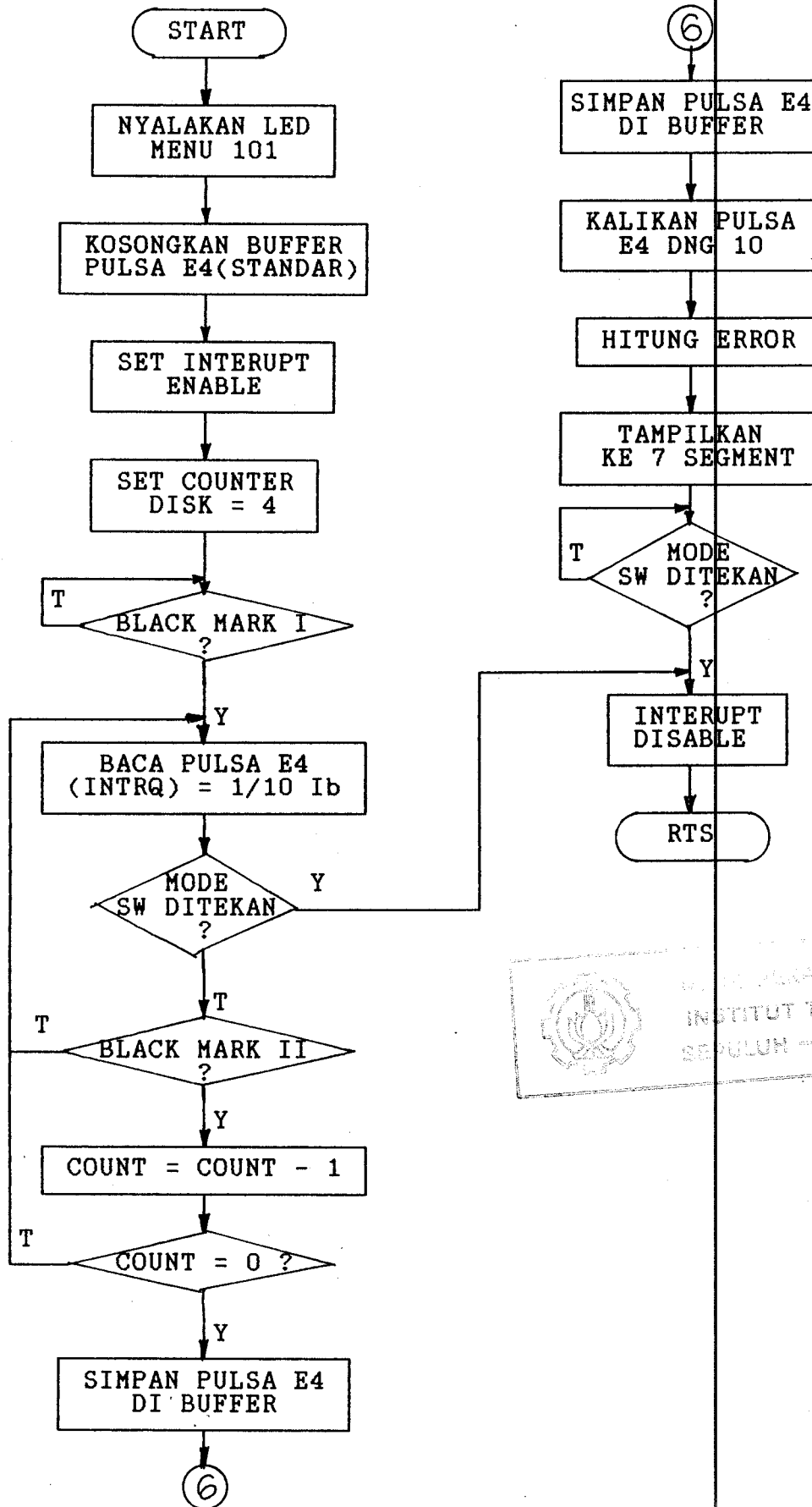


## SUB ROUTINE 2 Ib

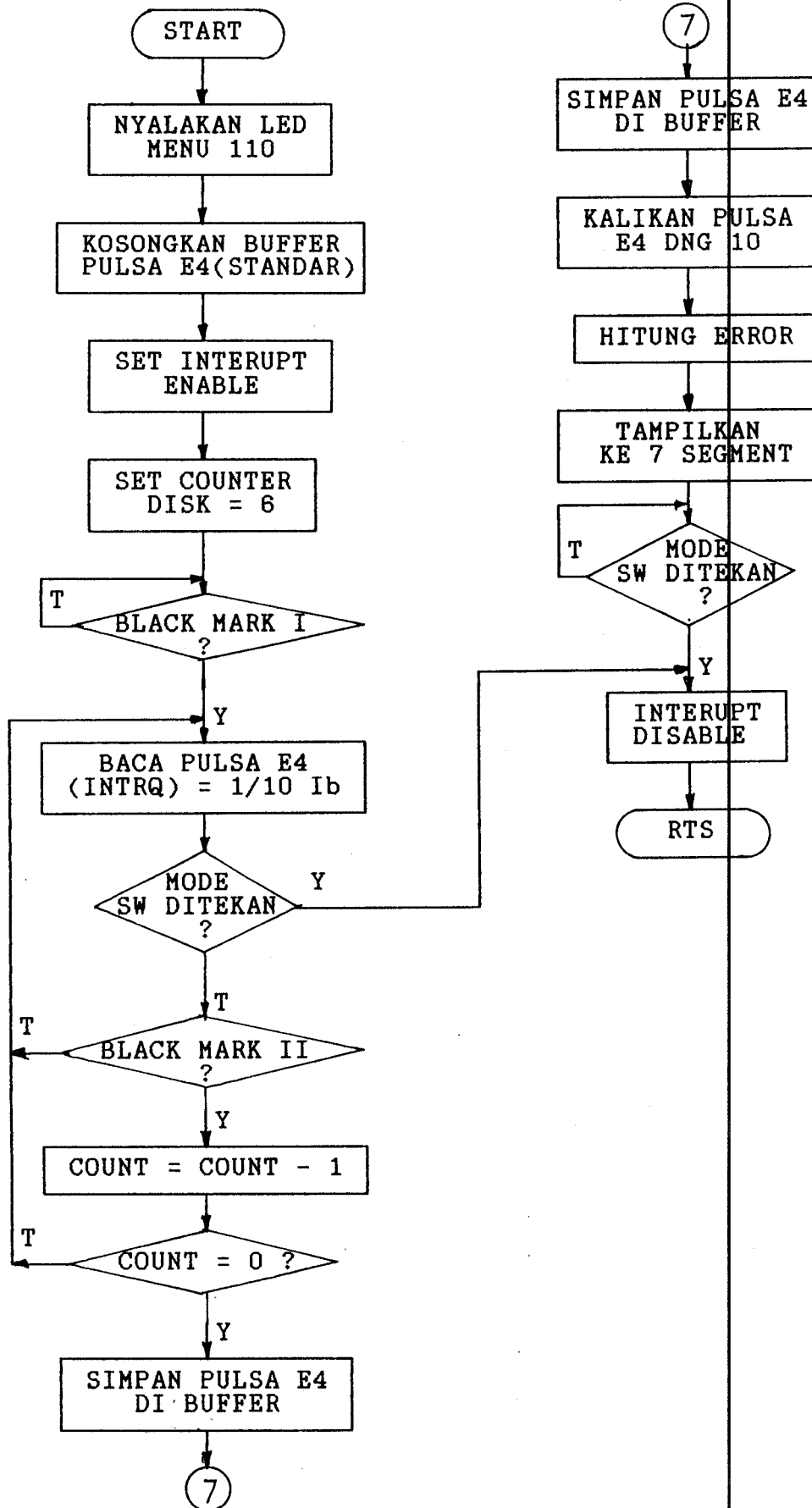




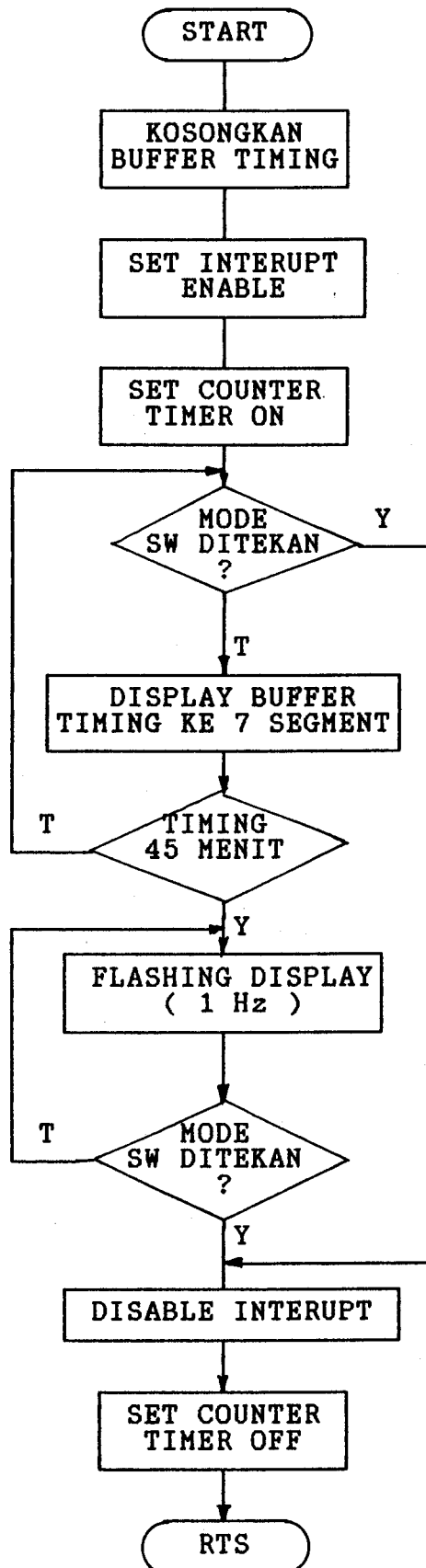
## SUB ROUTINE 4 Ib



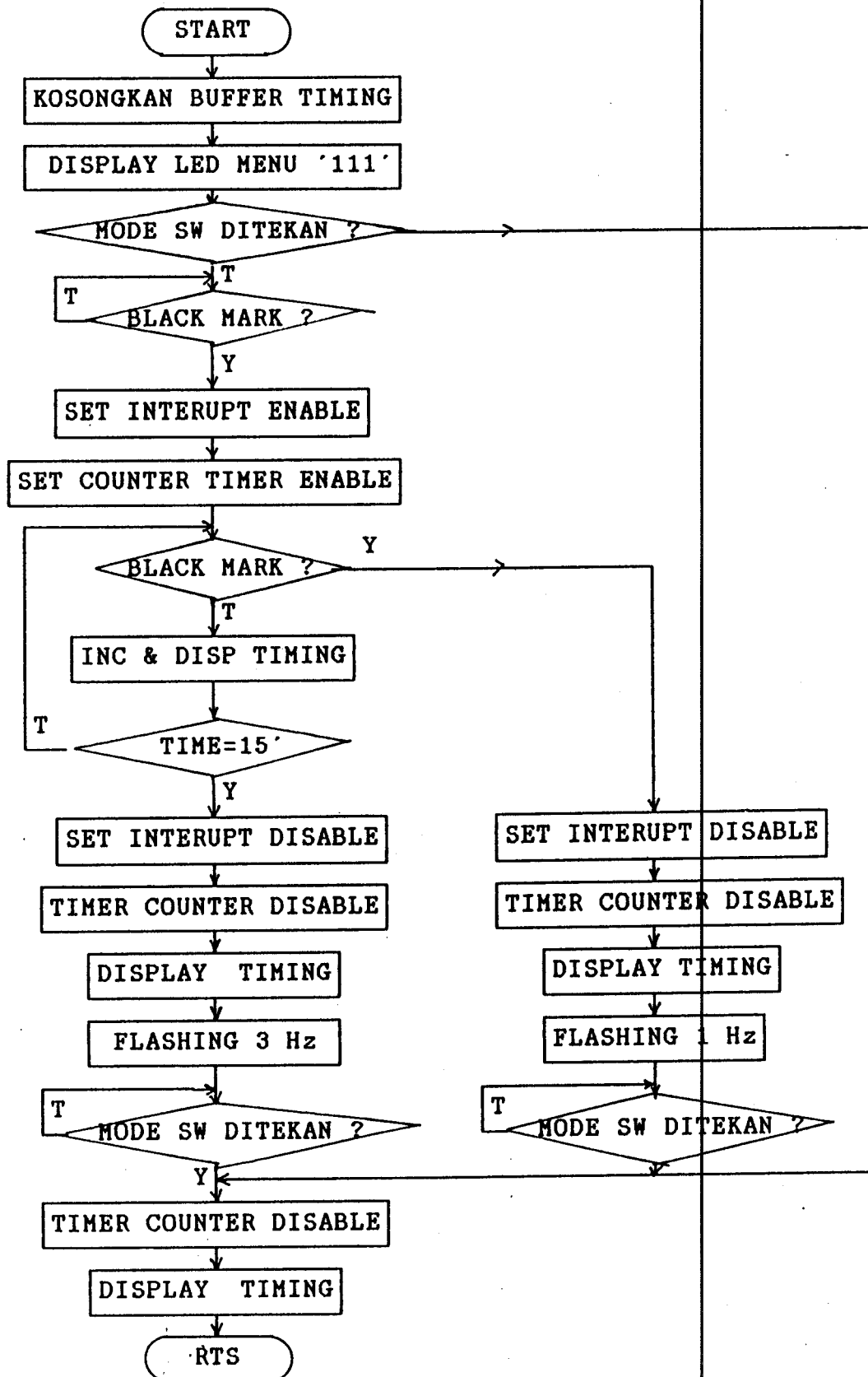
SUB ROUTINE 6 Ib



SUB ROUTINE WARMING UP



## SUB ROUTINE STARTING



## Technical Summary

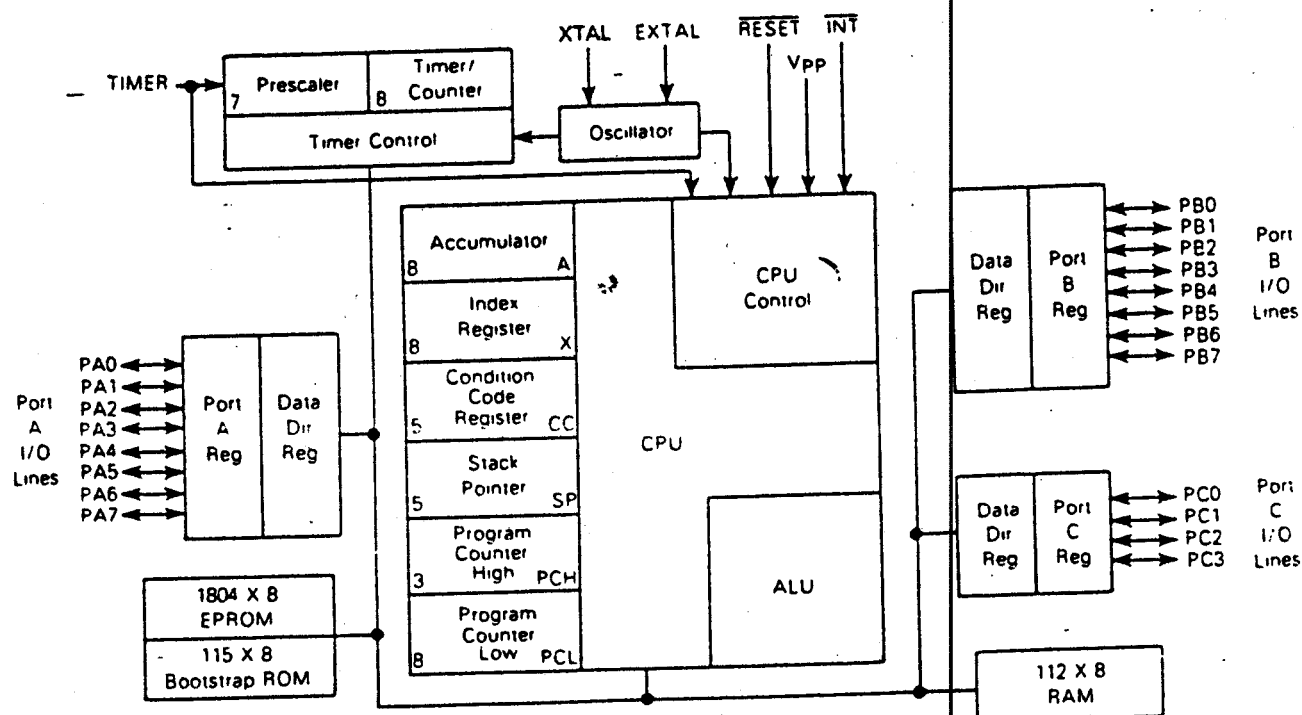
# 8-Bit EPROM Microcomputer Unit

The MC68705P3 (High-Density NMOS) Microcomputer Unit (MCU) is an EPROM member of the MC6805 Family of microcomputers. The user programmable EPROM allows program changes and lower volume applications. This low cost MCU has parallel I/O capability with pins programmable as input or output. This publication contains condensed information on the MCU; for detailed information, refer to *M6805 HMOS, M146805 CMOS Family User's Manual (M6805UM(AD2))* or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Internal 8-Bit Timer with 7-Bit Programmable Prescaler
- On-chip Oscillator
- Memory Mapped I/O
- Versatile Interrupt Handling
- Bit Manipulation
- Bit Test and Branch Instruction
- Vectored Interrupts
- Bootstrap program in ROM
- 1804 Bytes EPROM
- 112 Bytes RAM
- 20 TTL/CMOS Compatible Bidirectional I/O Lines

**BLOCK DIAGRAM**



This document contains information on a new product. Specifications and information herein are subject to change without notice.

## SIGNAL DESCRIPTION

### VCC AND VSS

Power is supplied to the microcomputer using these two pins. VCC is +5.25 volts ( $\pm 0.5\Delta$ ) power, and VSS is ground.

### VPP

This pin is used when programming the EPROM. In normal operation, this pin is connected to VCC.

### INT

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for more detailed information.

### EXTAL, XTAL

These pins provide control input for the on-chip clock oscillator circuit. A crystal, a resistor/capacitor combination, or an external signal is connected to these pins to provide a system clock. Selection is made by the CLK bit in the mask option register.

#### RC Oscillator

With this option, a resistor is connected to the oscillator pins as shown in Figure 1. The relationship between R and  $f_{osc}$  is shown in Figure 2.

#### Crystal

The circuit shown in Figure 1 is recommended when using a crystal. Using an external CMOS oscillator is recommended when crystals outside the specified ranges

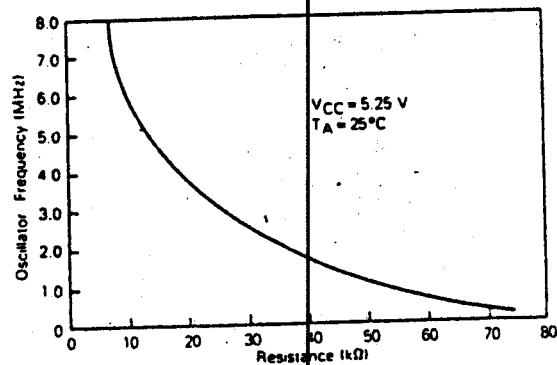


Figure 2. Typical Frequency vs Resistance for RC Oscillator Option only

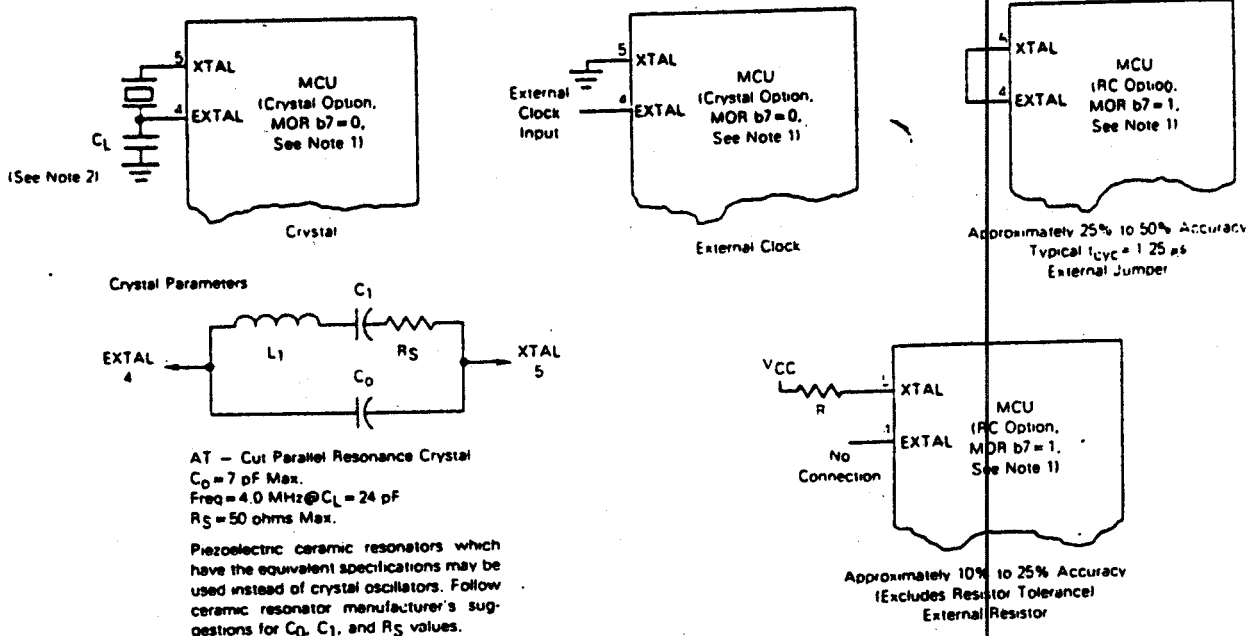
are to be used. The crystal and components should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time.

#### External Clock

An external clock should be applied to the EXTAL input with the XTAL input connected to ground, as shown in Figure 1. This option may only be used with the crystal oscillator option selected in the mask option register.

#### TIMER

This pin is used as an external input to control the internal timer/counter circuitry. This pin also detects a



#### NOTES:

- When the TIMER input pin is in the V<sub>HTP</sub> range (in the bootstrap EPROM programming mode), the crystal option is forced. When the TIMER input is at or below V<sub>CC</sub>, the clock generator option is determined by bit 7 of the mask option register (CLK).
- The recommended  $C_L$  value with a 4.0 MHz crystal is 27 pF maximum, including system distributed capacitance. There is an internal capacitance of approximately 25 pF on the XTAL pin. For crystal frequencies other than 4 MHz, the total capacitance on each pin should be scaled as the inverse of the frequency ratio. For example, with a 2 MHz crystal, use approximately 50 pF on EXTAL and approximately 25 pF on XTAL. The exact value depends on the motional-arm parameters of the crystal used.

Figure 1. Oscillator Connections

higher voltage level used to initiate the bootstrap program.

### RESET

This pin has a Schmitt trigger input and an on-chip pullup. The MCU can be reset by pulling RESET low. Refer to RESETS section for more detail.

### INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to PROGRAMMING for additional information.

## PROGRAMMING

### INPUT/OUTPUT PROGRAMMING

Any port pin is programmable as either input or output under software control of the corresponding write-only data direction register (DDR); DDRs always read "1". The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic 1 for output and a logic 0 for input. On reset, all the DDRs are initialized to a logic 0 state to put the ports in the input mode. The

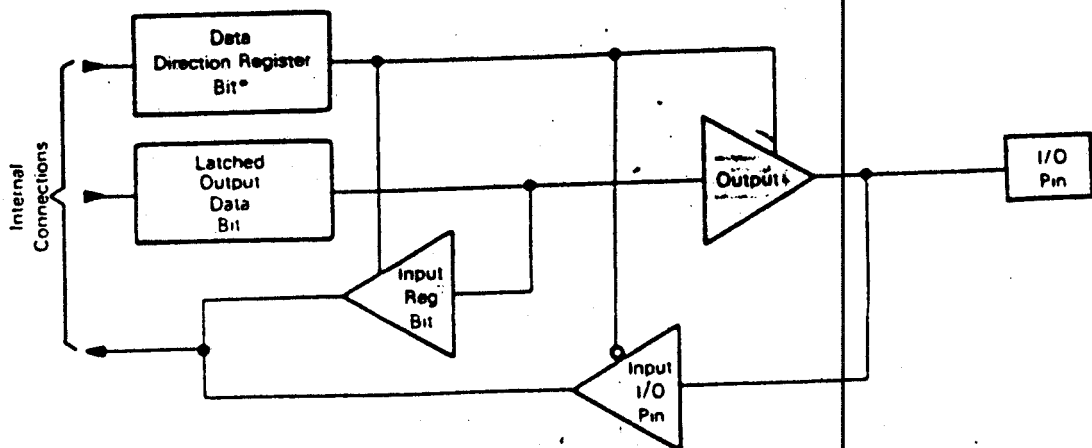
port output registers are not initialized on reset and should be written to before setting the DDR bits.

When programmed as outputs, the latched output data is readable as input data regardless of the logic levels at the output pin due to output loading. The latched output data bit may always be written. Therefore, any write to a port writes all of its data bits, even though the port DDR is set to input. This port write may be used to initialize the data registers and avoid undefined outputs. Care must be exercised when using read-modify-write instructions since the data read corresponds to the pin level if the DDR is an input (0) and also the latched output when the DDR is an output (1). Refer to Table 1 for I/O functions and to Figure 3 for typical port circuitry.

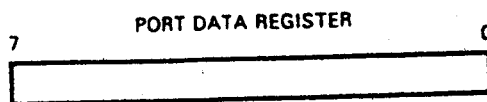
Table 1. I/O Pin Functions

Data Direction Register Bit	Output Data Bit	Output State	Input To MCU
1	0	0	0
1	1	1	1
0	X	Hi-Z**	Pin

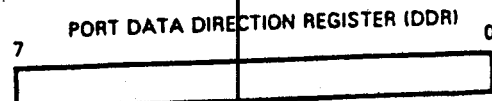
\*\*Ports A (with CMOS drive disabled), B, and C are three state ports. Port A has optional internal pullup devices to provide CMOS drive capability. See Electrical Characteristic tables for complete information.



\* DDR is a write-only register and reads as all "1s".



Port A Addr = \$000  
Port B Addr = \$001  
Port C Addr = \$002 (Bits 0—3)



(1) Write Only; reads as all "1s"  
(2) 1 = Output; 0 = Input. Cleared to 0 by reset.  
(3) Port A Addr = \$004  
Port B Addr = \$005  
Port C Addr = \$006 (Bits 0—3)

Figure 3. Typical Port I/O Circuitry and Register Configuration

## MEMORY

The MCU is capable of addressing 2048 bytes of memory and I/O registers. The memory map is shown in Figure 4. The locations consist of user EPROM, bootstrap ROM, RAM, a mask option register (MOR), a program control register, and I/O. The interrupt vectors are located from \$7F8 to \$7FF. The bootstrap is a mask-programmed ROM that allows the MCU to program its own EPROM.

The stack area is used during processing of an interrupt or subroutine call to save the CPU state. The stack pointer decrements during pushes and increments during pulls. Refer to **INTERRUPTS** for additional information.

## NOTE

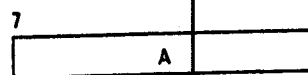
Using the stack area for data storage or temporary work locations requires care to prevent it from being overwritten due to stacking from an interrupt or subroutine call.

## REGISTERS

The MCU contains the registers described in the following paragraphs.

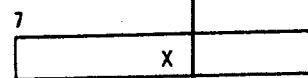
## ACCUMULATOR (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.



## INDEX REGISTER (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit value that may be added to an 8- or 16-bit immediate value to create an effective address. The index register may also be used as a temporary storage area.



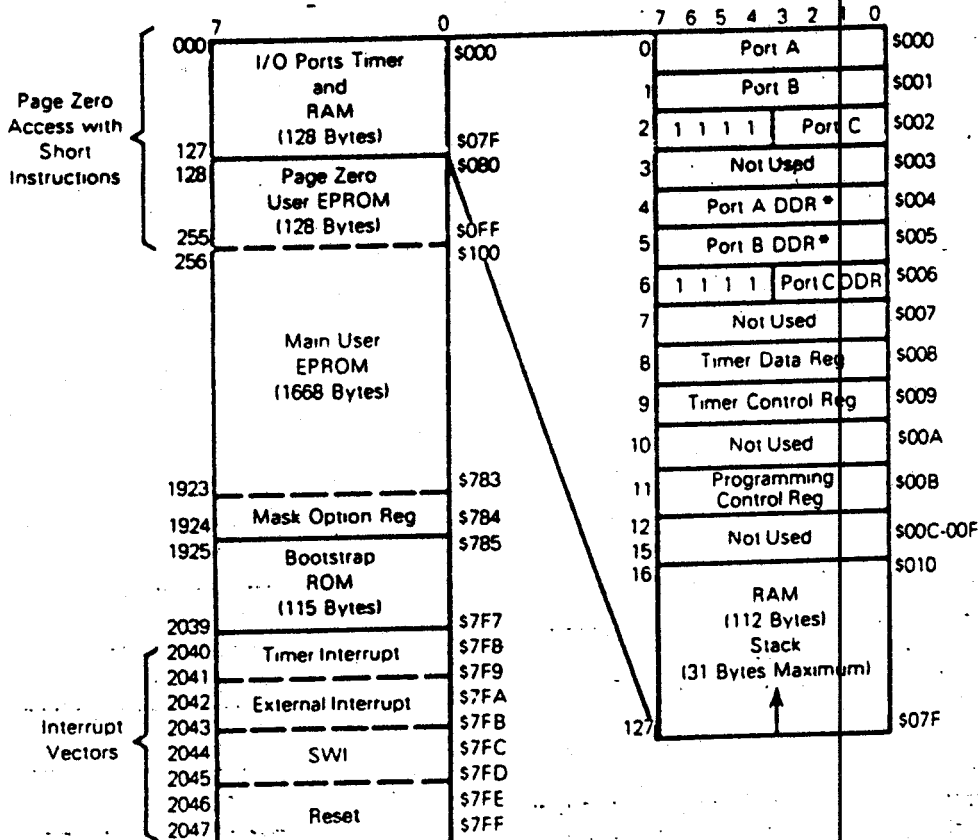
## PROGRAM COUNTER (PC)

The program counter is an 11-bit register that contains the address of the next byte to be fetched.



## STACK POINTER (SP)

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set at location \$07F. The stack pointer is then decremented as data is pushed onto the stack and incremented as data is pulled from the stack.

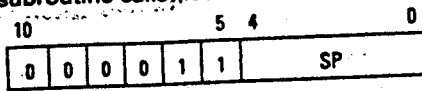


Caution: Data Direction Registers (DDRs) are write-only; they read as \$FF.

Figure 4. Memory Map



The six most-significant bits of the stack pointer are permanently set at 000011. Subroutines and interrupts may be nested down to location \$061 (31 bytes maximum), which allows the programmer to use up to 15 levels of subroutine calls (less if interrupts are allowed).



### CONDITION CODE REGISTER (CC)

The condition code register is a 5-bit register in which four bits are used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



#### Half Carry (H)

This bit is set during ADD and ADC operations to indicate that a carry occurred between bits 3 and 4.

#### Interrupt (I)

When this bit is set, the timer and external interrupt is masked (disabled). If an external interrupt occurs while this bit is set, the interrupt is latched and is processed as soon as the interrupt bit is cleared.

#### Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (bit 7 in the result is a logic 1).

#### Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

#### Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, and during shifts and rotates.

### RESETS

The MCU can be reset two ways: by initial power-up and by the external reset input (RESET). The RESET input consists mainly of a Schmitt trigger that senses the RESET line logic level.

#### POWER-ON-RESET (POR)

An internal reset is generated on power-up that allows the internal clock generator to stabilize. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. A delay of 100 milliseconds is required before

allowing the RESET input to go high. Connecting a capacitor to the RESET input (Figure 5) typically provides sufficient delay.

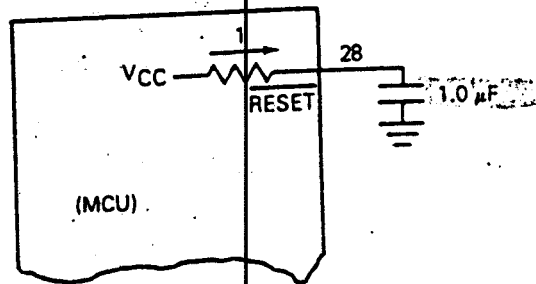


Figure 5. Power-up RESET Delay Circuit.

### EXTERNAL RESET INPUT

The MCU is reset when a logic zero is applied to the RESET input for a period longer than one machine cycle ( $t_{cyc}$ ). Under this type of reset, the Schmitt trigger switches off at  $V_{RES}$  to provide an internal reset voltage.

### INTERRUPTS

The MCU can be interrupted three different ways: (1) through the external interrupt INT input pin, (2) with the internal timer interrupt request, or (3) using the software interrupt instruction (SWI).

Interrupts cause the processor registers to be saved on the stack and the interrupt mask (1 bit) set to prevent additional interrupts. The RTI instruction causes the register contents to be recovered from the stack and then normal processing resumes. The stacking order is shown in Figure 6.

Unlike RESET, hardware interrupts do not cause the current instruction execution to be halted but are considered pending until the current instruction is complete.

#### NOTE

The current instruction is considered to be the one already fetched and being operated on.

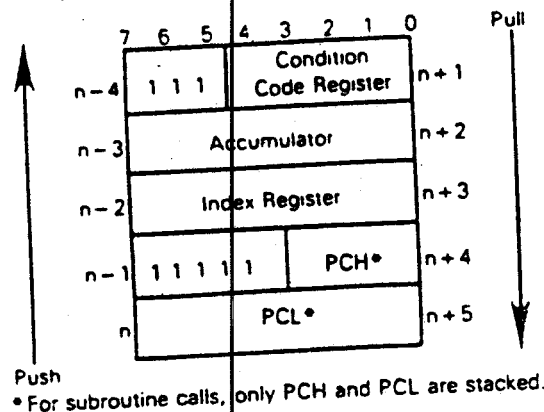


Figure 6. Interrupt Stacking Order

When the current instruction is complete, the processor checks all pending hardware interrupts and, if unmasked (I bit clear), proceeds with interrupt processing; otherwise, the next instruction is fetched and executed. Masked interrupts are latched for later interrupt service. If the timer interrupt status bit is cleared before unmasking the interrupt, then the interrupt is not latched.

If both an external interrupt and a timer interrupt are pending at the end of an instruction execution, the external interrupt is serviced first. The SWI is executed the same as any other instruction regardless of the setting of the I bit. Refer to Figure 7 for the reset and interrupt instruction processing sequence.

### TIMER INTERRUPT

If the timer mask bit (TCR6) is cleared, then, each time the timer decrements to zero (transitions from \$01 to \$00),

an interrupt request is generated. The actual processor interrupt is generated only if the interrupt mask bit of the condition code register (CCR) is also cleared. When the interrupt is recognized, the current state of the machine is pushed onto the stack, and the I bit in the CCR is set, masking further interrupts until the present one is serviced. The contents of the timer interrupt vector, containing the location of the timer interrupt service routine, is then loaded into the program counter. At the end of the timer interrupt service routine, the software normally executes an RTI instruction which restores the machine state and starts executing the interrupted program.

### EXTERNAL INTERRUPT

The external interrupt is internally synchronized and then latched on the falling edge of INT. Clearing the I bit enables the external interrupt. The following paragraphs describe two typical external interrupt circuits.

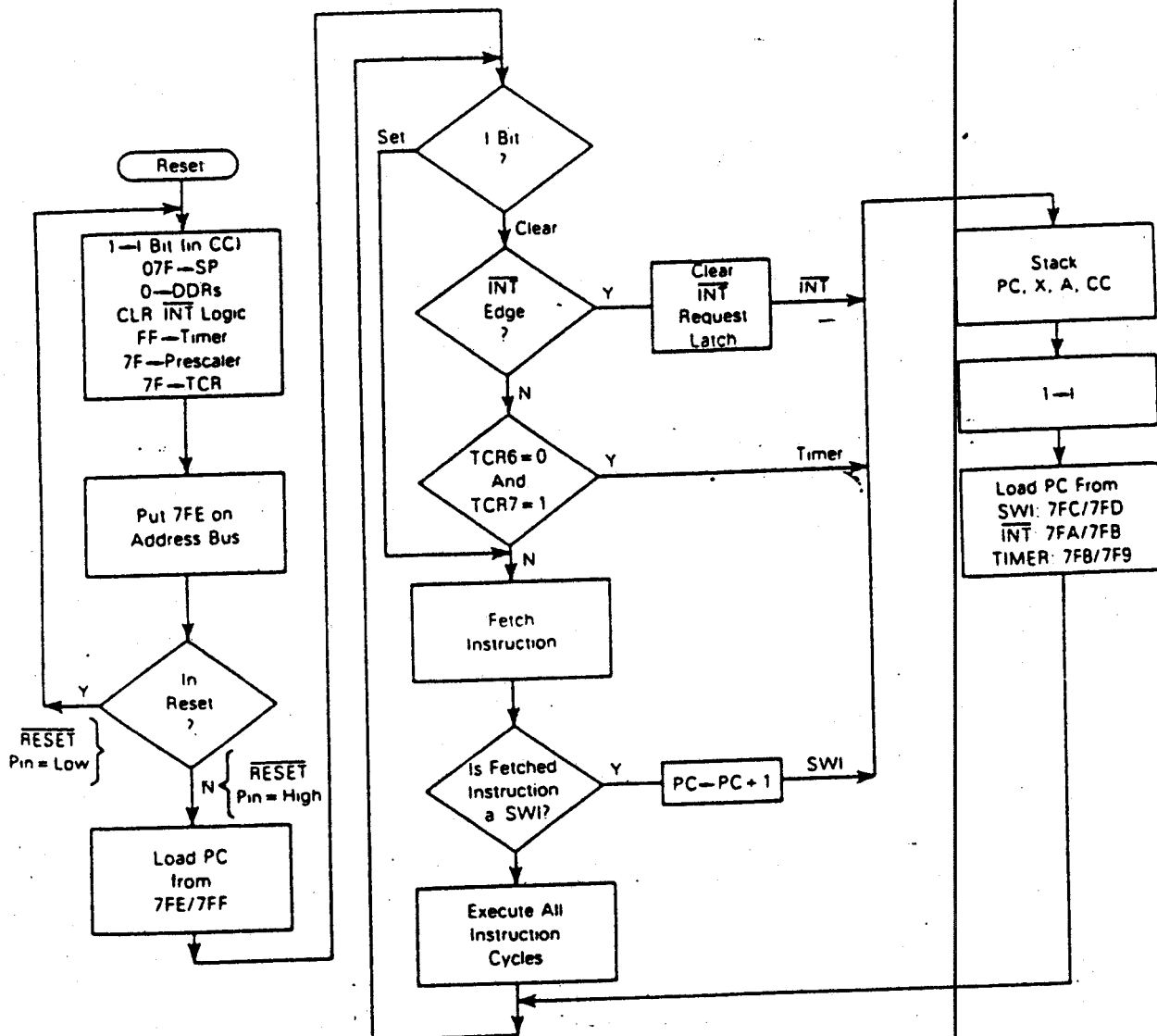


Figure 7. Reset and Interrupt Processing Flowchart

### Zero-Crossing

A sinusoidal input signal ( $f_{INT}$  maximum) can be used to generate an external interrupt (see Figure 8a) for use as a zero-crossing detector (for negative transitions of the ac sinusoid). This type of circuit allows applications such as servicing time-of-day routines and engaging/disengaging ac power control devices. Off-chip, full-wave rectification provides an interrupt at every zero crossing of the ac signal and thereby provides a 2f clock.

### Digital-Signal Interrupt

With this type of circuit (Figure 8b), the  $\overline{INT}$  pin can be driven by a digital signal. The maximum frequency of a signal that can be recognized by the TIMER or INT pin logic is dependent on the parameter labeled  $t_{WL}$ ,  $t_{WH}$ . Refer to TIMER for additional information.

### SOFTWARE INTERRUPT (SWI)

The SWI is an executable instruction that is executed regardless of the state of the I bit in the CCR. If the I bit

is zero, SWI executes after the other interrupts. The SWI execution is similar to the hardware interrupts.

### TIMER

The MCU consists of an 8-bit software programmable counter driven by a 7-bit software programmable prescaler. Various timer sources are made via the timer control register (TCR). The 8-bit counter may be loaded under program control and is decremented toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the timer control register (TCR) is set. Refer to Figure 9 for timer block diagram.

Timer interrupt can be masked (disabled) by setting the timer interrupt mask bit (bit 6) in the TCR. When the I bit in the condition code register is cleared, and TCR bit 6 is cleared, the processor receives the interrupt. The MCU responds to this interrupt by 1) saving the present CPU state on the stack, 2) fetching the timer interrupt vector,

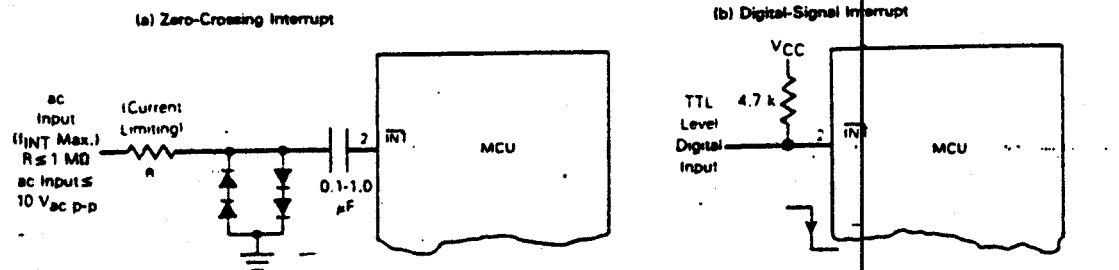


Figure 8. Typical Interrupt Circuits

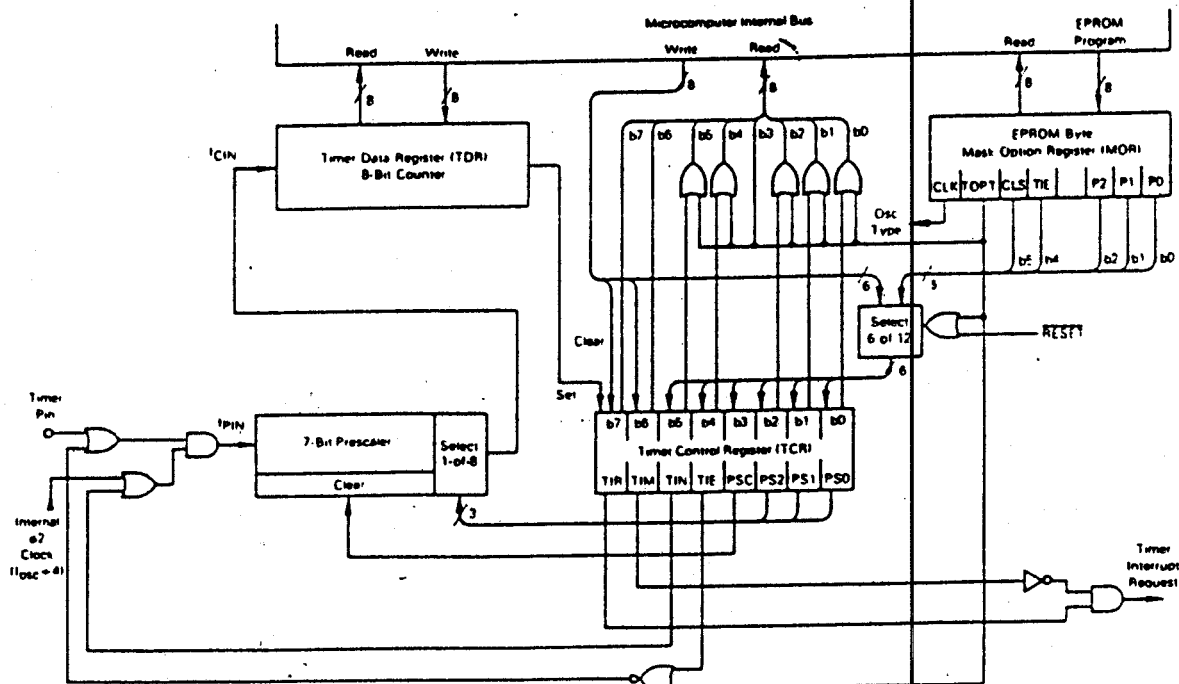


Figure 9. Timer Block Diagram

and 3) executing the interrupt routine. Timer interrupt request bit must be cleared by software. Refer to **RESETS and INTERRUPTS** for additional information.

The prescaler is a 7-bit divider which is used to extend the maximum length of the timer. To avoid truncation errors, the prescaler is cleared when TCR bit 3 is set to a logic 1; however, TCR bit 3 always reads as a logic 0 to ensure proper operation with read-modify-write instructions.

The timer continues to count past zero, falling from \$00 through \$FF, and continues the countdown. The counter can be read at any time by reading the timer data register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process. TDR is unaffected by reset.

### SOFTWARE CONTROLLED MODE

The timer prescaler input can be configured for three different operating modes plus a disable mode, depending on the value written to TCR control bits 4 and 5 (TIE and TIN). The following paragraphs describe the different modes.

#### Timer Input Mode 1

When TIE and TIN are both programmed to zero, the timer input is from the internal clock (phase 2) and the timer input pin is disabled. The internal clock mode can be used for periodic interrupt generation as well as a reference for frequency and event measurement.

#### Timer Input Mode 2

When TIE = 1 and TIN = 0, the internal clock and the timer input signals are ANDed to form the timer input. This mode can be used to measure external pulse widths. The active high, external pulse gates in the internal clock for the duration of the external pulse. The accuracy of the count is  $\pm 1$ .

#### Timer Input Mode 3

When TIE = 0 and TIN = 1, no prescaler input frequency is applied to the prescaler and the timer is disabled.

#### Timer Input Mode 4

When TIE and TIN are both one, the timer input is from the external clock. The external clock can be used to count external events as well as to provide an external frequency for generating periodic interrupts.

### MOR CONTROLLED MODE

This mode is selected when TOPT (bit 6) in the MOR is programmed to logic 1. The timer circuits are the same as described in **SOFTWARE CONTROLLED MODE**. The logic levels of TCR bits 0, 1, 2, and 5 are determined during EPROM programming by the same bits in the MOR. Therefore, bits 0, 1, 2, and 5 in the MOR control the prescaler division and the timer clock selection. TIE (bit 4) and PSC (bit 3) in the TCR are set to a logic 1 when in the MOR controlled mode. TIM (bit 6) and TIR (bit 7) are controlled by the counter and software.

### TIMER CONTROL REGISTER (TCR) \$009

This is an 8-bit register that controls various functions such as configuring operation mode, setting ratio of the prescaler, and generating timer interrupt request signal. All bits are read/write except bit 3. When the MOR<sub>TOPT</sub> = 1, then bits 5, 2, 1, and 0 in the TCR take on the corresponding bits of the MOR during reset.

7	6	5	4	3	2	1	0
TIR	TIM	1	1	1	1	1	1

RESET:

0

1

U

J

TCR with MOR<sub>TOPT</sub> = 1 (MC6805P2/P6 Emulation)

7	6	5	4	3	2	1	0
TIR	TIM	TIN	TIE	PSC	PS2	PS1	PS0

RESET:

0

1

TCR with MOR<sub>TOPT</sub> = 0 (Software Programmable Timer)

TIR — Timer Interrupt Request

Used to indicate the timer interrupt when it is logic one

1 = Set when the timer data register changes to all zeros

0 = Cleared by external reset, power-on reset, or under program control

TIM — Timer Interrupt Mask

Used to inhibit the timer interrupt.

1 = Interrupt inhibited

0 = Interrupt enabled

TIN — External or Internal

Selects input clock source

1 = External clock selected

0 = Internal clock selected ( $f_{osc}/4$ )

TIE — TIMER External Enable

Used to enable external TIMER pin

1 = Enables external timer pin

0 = Disables external timer pin

PSC — Prescaler Clear

Write only bit. Writing a 1 to this bit resets the prescaler to zero. A read of this location always indicates a zero.

PS2, PS1, PS0 — Prescaler Select Bits

Decoded to select one of eight outputs of the prescaler

PS2	PS1	PS0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

**MASK OPTION REGISTER (MOR)**

The MOR is implemented in EPROM and contains all zeros prior to programming. This register is not affected by reset. The MOR bits are described in the following paragraphs.

7	6	5	4	3	2	1	0
CLK	TOPT	CLS	TIE		P2	P1	P0

CLK — Clock (oscillator type)

1 = Resistor Capacitor (RC)

0 = Crystal

TOPT — Timer Option

1 = MC6805P2/P6 type timer/prescaler. All bits except 6 and 7 of the TCR are invisible to the user. Bits 5, 2, 1, and 0 of the MOR determine the equivalent MC6805P2/P6 mask options.

0 = All TCR bits are implemented as a software programmable timer. The state of MOR bits 5, 4, 2, 1, and 0 sets the initial values of their respective TCR bits.

CLS — Timer/Prescaler Clock Source

1 = External TIMER pin

0 = Internal clock

TIE — Timer External Enable

Not used if TOPT = 1. Sets the initial value of TIE in the TCR if TOPT = 0.

1 = Not used

0 = Sets initial value of TIE in the TCR

P2, P1, P0

The logical levels of these bits, when decoded, select one of eight outputs on the timer prescaler.

P2	P1	P0	Prescaler Division
0	0	0	1 (Bypass Prescaler)
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

**PROGRAMMING CONTROL REGISTER (PCR)**

The PCR is an 8-bit register which provides the necessary control bits to program the EPROM. The bootstrap program manipulates the PCR when programming, so the user need not be concerned with PCR in most applications.

7	6	5	4	3	2	1	0
1	1	1	1	1	VPON	PGE	PLE

RESET:

U U U U U U 1 1

PLE — Programming Latch Enable

Controls address and data being latched into the EPROM. Set during reset, but may be cleared any time.

1 = Read EPROM

0 = Latch address and data on EPROM

PGE — Program Enable

Enables programming of EPROM. Must be set when changing the address and data. Set during reset.

1 = Inhibit EPROM programming

0 = Enable EPROM programming (if PLE is low)

VPON — Vpp On

A read-only bit that indicates high voltage at the Vpp pin. When set to "1", disconnects PGE and PLE from the chip.

1 = No high voltage on Vpp pin

0 = High voltage on Vpp pin

**NOTE**

VPON being "0" does not indicate that the Vpp level is correct for programming. It is used as a safety interlock for the user in the normal operating mode.

VPON	PGE	PLE	Programming Conditions
0	0	0	Programming Mode (Program EPROM Byte)
1	0	0	PGE and PLE Disabled from System
0	1	0	Programming Disabled (Latch Address and Data in EPROM)
1	1	0	PGE and PLE Disabled from System
0	0	1	Invalid State: PGE = 0 if PLE = 0
1	0	1	Invalid State: PGE = 0 if PLE = 0
0	1	1	"High Voltage" on Vpp
1	1	1	PGE and PLE Disabled from System (Operating Mode)

**EPROM PROGRAMMING****PROGRAMMING**

The MCU bootstrap program can be used to program the MCU EPROM.

A 2764 UV EPROM must first be programmed with the same information that is to be transferred to the MCU EPROM. Refer to application note, MC68705P3/R3/U3 8-bit EPROM Microcomputer Programming Module (AN-857 Rev 2) for a schematic diagram and instructions on programming the MCU EPROM.

**EMULATION**

The MC68705P3 emulates the MC6805P2 and MC6805P6 "exactly." The MC6805P2/P6 mask features are implemented in the mask-option register (MOR) EPROM byte on the MC68705P3. A few minor exceptions to the exactness of emulation are listed below:

1. The MC68705P2/P6 "future ROM" area is implemented in the MC68705P3, and these 704 bytes must

be left unprogrammed to accurately simulate the MC6805P2/P6. The MC6805P2/P6 read all "0s" from this area.

- The reserved ROM areas in the MC6805P2/P6 and the MC68705P3 have different data stored in them. This data is subject to change without notice. The MC6805P2/P6 use the reserved ROM for the self-check feature, and the MC68705P3 uses this area for the bootstrap program.
- The MC6805P2/P6 read all "1s" in its 48-byte "future RAM" area. This RAM is not implemented in the MC6805P2/P6 mask ROM versions but is implemented in the MC68705P3.
- The Vpp line (pin 6) in the MC68705P3 must be tied to VCC for normal operation. In the MC6805P2/P6, pin 6 is the NUM pin and is grounded in normal operation.
- The LVI feature is not available in the MC68705P3. Processing differences are not presently compatible with proper design of this feature in the EPROM version.

The operation of all other circuitry has been exactly duplicated or designed to function identically in both devices including interrupts, timer, data ports, and data direction registers (DDRs). A design goal has been to provide the user with a safe, inexpensive way to verify a program and system design before committing to a factory programmed ROM.

## INSTRUCTION SET

The MCU has a set of 59 basic instructions which can be divided into five different types: register/memory, read-modify-write, branch, bit manipulation, and control. The following paragraphs briefly explain each type.

### REGISTER/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to the following list of instructions.

Function	Mnemonic
Load A from Memory	LDA
Load X from Memory	LDX
Store A in Memory	STA
Store X in Memory	STX
Add Memory to A	ADD
Add Memory and Carry to A	ADC
Subtract Memory	SUB

— Continued —

Function	Mnemonic
Subtract Memory from A with Borrow	SBC
AND Memory to A	AND
OR Memory with A	ORA
Exclusive OR Memory with A	EOR
Arithmetic Compare A with Memory	CMP
Arithmetic Compare X with Memory	CPX
Bit Test Memory with A (Logical Compare)	BIT
Jump Unconditional	JMP
Jump to Subroutine	JSR

### READ-MODIFY-WRITE INSTRUCTIONS

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read-modify-write sequence since it does not modify the value. Refer to the following list of instructions.

Function	Mnemonic
Increment	INC
Decrement	DEC
Clear	CLR
Complement	COM
Negate (2's Complement)	NEG
Rotate Left Thru Carry	ROL
Rotate Right Thru Carry	ROR
Logical Shift Left	LSL
Logical Shift Right	LSR
Arithmetic Shift Right	ASR
Test for Negative or Zero	TST

### BRANCH INSTRUCTIONS

This set of instructions branches if a particular condition is met; otherwise, no operation is performed. Branch instructions are two byte instructions. Refer to the following list for branch instructions.

Function	Mnemonic
Branch Always	BRA
Branch Never	BRN
Branch if Higher	BHI
Branch if Lower or Same	BLS
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)

— Continued —

Function	Mnemonic
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if Not Equal	BNE
Branch if Equal	BEQ
Branch if Half Carry Clear	BHCC
Branch if Half Carry Set	BHCS
Branch if Plus	BPL
Branch if Minus	BMI
Branch if Interrupt Mask Bit is Clear	BMC
Branch if Interrupt Mask Bit is Set	BMS
Branch if Interrupt Line is Low	BIL
Branch if Interrupt Line is High	BIH
Branch to Subroutine	BSR

### CONTROL INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for control instructions.

Function	Mnemonic
Transfer A to X	TAX
Transfer X to A	TXA
Set Carry Bit	SEC
Clear Carry Bit	CLC
Set Interrupt Mask Bit	SEI
Clear Interrupt Mask Bit	CLI
Software Interrupt	SWI
Return from Subroutine	RTS
Return from Interrupt	RTI
Reset Stack Pointer	RSP
No-Operation	NOP

### BIT MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit which resides in the first 256 bytes of the memory space, where all port registers, port DDRs, timer, timer control, and on-chip RAM reside. An additional feature allows the software to test and branch on the state of any bit within these 256 locations. The bit set, bit clear and bit test, and branch functions are all implemented with a single instruction. For test and branch instructions, the value of the bit tested is also placed in the carry bit of the condition

code register. Refer to the following list for bit manipulation instructions.

Function	Mnemonic
Branch if Bit n is Set	BRSET n (n=0...7)
Branch if Bit n is Clear	BRCLR n (n=0...7)
Set Bit n	BSET n (n=0...7)
Clear Bit n	BCLR n (n=0...7)

### OPCODE MAP SUMMARY

Table 2 is an opcode map for the instructions used on the MCU.

### ADDRESSING MODES

The MCU uses ten different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. The various indexed addressing modes make it possible to locate data tables, code conversion tables, and scaling tables anywhere in the memory space. Short indexed accesses are single byte instructions, while the longest instructions (three bytes) permit accessing tables throughout memory. Short and long absolute addressing is also included. Two byte direct addressing instructions access all data bytes in most applications. Extended addressing permits jump instructions to reach all memory.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored.

#### IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. The immediate addressing mode is used to access constants that do not change during program execution (e.g., a constant used to initialize a loop counter).

#### DIRECT

In the direct addressing mode, the effective address of the argument is contained in a single byte following the opcode byte. Direct addressing allows the user to directly address the lowest 256 bytes in memory with a single two-byte instruction.

#### EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. Instructions with extended addressing mode are capable of referencing arguments anywhere in memory with a single three-byte instruction.

#### RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true.

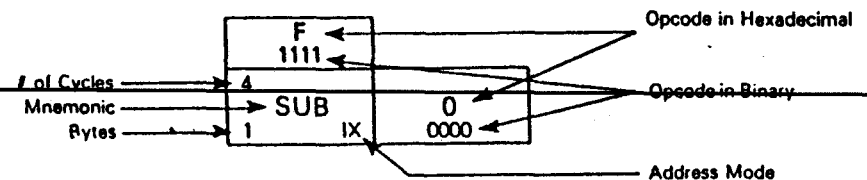
Table 2. Opcode Map

Low		Bit Manipulation		Branch	Read-Modify-Write				Control		Register/Memory						High		
		BTB	BSC	REL	DIR	INH	INH	IX1	IX	INH	INH	IMM	DIR	EXT	IX2	IX1			IX
0		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	Low	
0		BRSET0 BTB	BSET0 BSC	BRA REL	NEG DIR	NEG INH	NEG INH	NEG IX1	NEG IX	RTI INH		SUB IMM	SUB DIR	SUB EXT	SUB IX2	SUB IX1	SUB IX	0	0000
1		BRCLR0 BTB	BCLR0 BSC	BRN REL						RTS INH		CMP IMM	CMP DIR	CMP EXT	CMP IX2	CMP IX1	CMP IX	1	0001
2		BRSET1 BTB	BSET1 BSC	BHI REL								SBC IMM	SBC DIR	SBC EXT	SBC IX2	SBC IX1	SBC IX	2	0010
3		BRCLR1 BTB	BCLR1 BSC	BLS REL	COM DIR	COMA INH	COMX INH	COM IX1	COM IX	SWI INH		CPX IMM	CPX DIR	CPX EXT	CPX IX2	CPX IX1	CPX IX	3	0011
4		BRSET2 BTB	BSET2 BSC	BCC REL	LSR DIR	LSRA INH	LSRX INH	LSR IX1	LSR IX			AND IMM	AND DIR	AND EXT	AND IX2	AND IX1	AND IX	4	0100
5		BRCLR2 BTB	BCLR2 BSC	BCS REL								BIT IMM	BIT DIR	BIT EXT	BIT IX2	BIT IX1	BIT IX	5	0101
6		BRSET3 BTB	BSET3 BSC	BNE REL	ROR DIR	RORA INH	RORX INH	ROR IX1	ROR IX			LDA IMM	LDA DIR	LDA EXT	LDA IX2	LDA IX1	LDA IX	6	0110
7		BRCLR3 BTB	BCLR3 BSC	BEO REL	ASR DIR	ASRA INH	ASRX INH	ASR IX1	ASR IX	TAX INH		STA IMM	STA DIR	STA EXT	STA IX2	STA IX1	STA IX	7	0111
8		BRSET4 BTB	BSET4 BSC	BHCC REL	LSL DIR	LSLA INH	LSLX INH	LSL IX1	LSL IX	CLC INH		EOR IMM	EOR DIR	EOR EXT	EOR IX2	EOR IX1	EOR IX	8	1000
9		BRCLR4 BTB	BCLR4 BSC	BHCS REL	ROL DIR	ROLA INH	ROLX INH	ROL IX1	ROL IX	SEC INH		ADC IMM	ADC DIR	ADC EXT	ADC IX2	ADC IX1	ADC IX	9	1001
A		BRSET5 BTB	BSET5 BSC	BPL REL	DEC DIR	DECA INH	DECX INH	DEC IX1	DEC IX	CLI INH		ORA IMM	ORA DIR	ORA EXT	ORA IX2	ORA IX1	ORA IX	A	1010
B		BRCLR5 BTB	BCLR5 BSC	BMI REL						SEI INH		ADD IMM	ADD DIR	ADD EXT	ADD IX2	ADD IX1	ADD IX	B	1011
C		BRSET6 BTB	BSET6 BSC	BMC REL	INC DIR	INCA INH	INCX INH	INC IX1	INC IX	RSP INH		JMP IMM	JMP DIR	JMP EXT	JMP IX2	JMP IX1	JMP IX	C	1100
D		BRCLR6 BTB	BCLR6 BSC	BMS REL	TST DIR	TSTA INH	TSTX INH	TST IX1	TST IX	NOP INH		BSR REL	JSR DIR	JSR EXT	JSR IX2	JSR IX1	JSR IX	D	1101
E		BRSET7 BTB	BSET7 BSC	BIL REL								LDX IMM	LDX DIR	LDX EXT	LDX IX2	LDX IX1	LDX IX	E	1110
F		BRCLR7 BTB	BCLR7 BSC	BIH REL	CLR DIR	CLRA INH	CLRX INH	CLR IX1	CLR IX	TXA INH		STX IMM	STX DIR	STX EXT	STX IX2	STX IX1	STX IX	F	1111

## Abbreviations for Address Modes

INH	Inherent
IMM	Immediate
DIR	Direct
EXT	Extended
REL	Relative
BSC	Bit Set/Clear
BTB	Bit Test and Branch
IX	Indexed (No Offset)
IX1	Indexed, 1 Byte (8-Bit) Offset
IX2	Indexed, 2 Byte (16-Bit) Offset

## LEGEND





Otherwise, control proceeds to the next instruction. The span of relative addressing is from -126 to +129 from the opcode address.

#### INDEX, NO OFFSET

In the indexed, no offset addressing mode, the effective address of the argument is contained in the 8-bit index register. This addressing mode can access the first 256 memory locations. These instructions are only one byte long. This mode is often used to move a pointer through a table or to hold the address of a frequently referenced RAM or I/O location.

#### INDEXED, 8-BIT OFFSET

In the indexed, 8-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the unsigned byte following the opcode. The addressing mode is useful for selecting the Kth element in an n element table. With this 2-byte instruction, K would typically be in X with the address of the beginning of the table in the instruction. As such, tables may begin anywhere within the first 256 addressable locations and could extend as far as location 510 (\$1FE is the last location at which the instruction may begin).

#### INDEXED, 16-BIT OFFSET

In the indexed, 16-bit offset addressing mode, the effective address is the sum of the contents of the unsigned 8-bit index register and the two unsigned bytes following the opcode. This addressing mode can be used in a manner similar to indexed, 8-bit offset except that this 3-byte instruction allows tables to be anywhere in memory.

#### BIT SET/CLEAR

In the bit set/clear addressing mode, the bit to be set or cleared is part of the opcode. The byte following the opcode specifies the direct addressing of the byte to which

the specified bit is to be set or cleared. Any read/write bit in the first 256 locations of memory, including I/O, can be selectively set or cleared with a single 2-byte instruction.

#### CAUTION

The corresponding DDRs for ports A, B, and C are write only registers (registers at \$004, \$005, and \$006). A read operation on these registers always returns "1". Since BSET and BCLR are read-modify-write functions, they cannot be used to set or clear a DDR bit (all "unaffected" bits would be set). It is recommended that all DDR bits in a port be written using a single-store instruction.

#### BIT TEST AND BRANCH

The bit test and branch addressing mode is a combination of direct addressing and relative addressing. The bit that is to be tested and its condition (set or clear) is included in the opcode. The address of the byte to be tested is in the single byte immediately following the opcode byte. The signed relative 8-bit offset in the third byte is added to the PC if the specified bit is set or cleared in the specified memory location. This single 3-byte instruction allows the program to branch based on the condition of any readable bit in the first 256 locations of memory. The span of branching is from -125 to +130 from the opcode address. The state of the tested bit is also transferred to the carry bit of the condition code register.

#### INHERENT

In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one byte long.

## ELECTRICAL SPECIFICATIONS

## MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V <sub>CC</sub>	-0.3 to +7.0	V
Input Voltages			
EPROM Programming Voltage (V <sub>pp</sub> Pin)	V <sub>pp</sub>	-0.3 to +22.0	V
TIMER Pin (Normal Mode)	V <sub>in</sub>	-0.3 to +7.0	V
TIMER Pin (Bootstrap Programming Mode)	V <sub>in</sub>	-0.3 to +15.0	V
All Others	V <sub>in</sub>	-0.3 to +7.0	V
Operating Temperature Range	T <sub>A</sub>	T <sub>L</sub> to T <sub>H</sub> 0 to +70	°C
Storage Temperature Range	T <sub>stg</sub>	-55 to +150	°C
Junction Temperature Cerdip	T <sub>J</sub>	150	°C/W

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. For proper operation, V<sub>in</sub> and V<sub>out</sub> should be constrained to the range V<sub>SS</sub> ≤ (V<sub>in</sub> or V<sub>out</sub>) ≤ V<sub>CC</sub>. Reliability of operation is enhanced if unused inputs except EXTAL are tied to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>CC</sub>).

## THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Cerdip	θ <sub>JA</sub>	60	°C/W



MILIK PERPUSTAKAAN  
INSTITUT TEKNOLOGI  
SEPULUH - NOPEMBER

## POWER CONSIDERATIONS

The average chip-junction temperature, T<sub>J</sub>, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T<sub>A</sub> = Ambient Temperature, °C

θ<sub>JA</sub> = Package Thermal Resistance,  
Junction-to-Ambient, °C/W

P<sub>D</sub> = P<sub>INT</sub> + P<sub>I/O</sub>

P<sub>INT</sub> = I<sub>CC</sub> × V<sub>CC</sub>, Watts - Chip Internal Power

P<sub>I/O</sub> = Power Dissipation on Input and Output  
Pins - User Determined

For most applications P<sub>I/O</sub> < P<sub>INT</sub> and can be neglected.

The following is an approximate relationship between P<sub>D</sub> and T<sub>J</sub> (if P<sub>I/O</sub> is neglected):

$$P_D = K + (T_J - 273^\circ\text{C}) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273^\circ\text{C}) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P<sub>D</sub> (at equilibrium) for a known T<sub>A</sub>. Using this value of K, the values of P<sub>D</sub> and T<sub>J</sub> can be obtained by solving equations (1) and (2) iteratively for any value of T<sub>A</sub>.

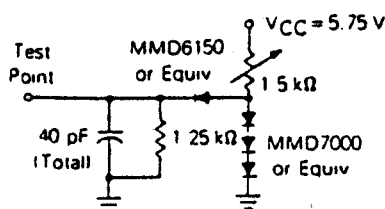


Figure 10. TTL Equivalent Test Load  
(Port B)

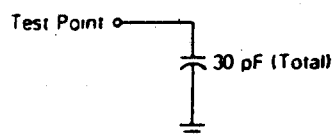


Figure 11. CMOS Equivalent Test Load  
(Port A)

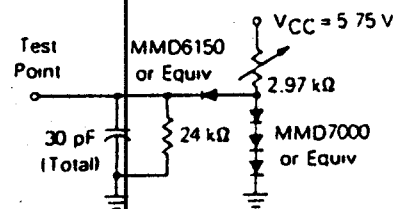


Figure 12. TTL Equivalent Test Load  
(Ports A and C)

ELECTRICAL CHARACTERISTICS ( $V_{CC}=5.25 \pm 0.5$  Vdc,  $V_{SS}=0$  Vdc,  $T_A=0^\circ\text{C}$  to  $70^\circ\text{C}$ , unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Input High Voltage RESET ( $4.75 \leq V_{CC} \leq 5.75$ ) INT ( $4.75 \leq V_{CC} \leq 5.75$ ) All Other	$V_{IH}$	4.0 $V_{CC} - 0.5$ 4.0 $V_{CC} - 0.5$ 2.0	— — ** —	$V_{CC}$ $V_{CC}$ $V_{CC}$ $V_{CC}$	V
Input High Voltage (TIMER Pin) Timer Mode Bootstrap Programming Mode	$V_{IH}$	2.0 9.0	— 12.0	$V_{CC}$ 15.0	V
Input Low Voltage RESET INT All Other	$V_{IL}$	-0.3 -0.3 -0.3	— ** —	0.8 1.5 0.8	V
Internal Power Dissipation (No Port Loading, $V_{CC}=5.25$ V, $T_A=0^\circ\text{C}$ )	$P_{INT}$	—	450	TBD	mW
Input Capacitance XTAL All Other	$C_{in}$	— —	25 10	— —	pF
INT Zero-Crossing Voltage, through a Capacitor	$V_{INT}$	2.0	—	4.0	$V_{acc-p}$
RESET Hysteresis Voltage Out of Reset Voltage Into Reset Voltage	$V_{IRES+}$ $V_{IRES-}$	2.1 0.8	— —	4.0 2.0	V
Programming Voltage ( $V_{pp}$ Pin) Programming EPROM Operating Mode	$V_{pp}^*$	20.0 4.0	21.0 $V_{CC}$	22.0 5.75	V
Input Current TIMER ( $V_{in}=0.4$ V) INT ( $V_{in}=0.4$ V) EXTAL ( $V_{in}=2.4$ V to $V_{CC}$ Crystal Option) ( $V_{in}=0.4$ V Crystal Option) RESET ( $V_{in}=0.8$ V) (External Capacitor Changing Current)	$I_{in}$	— — — — -4.0	— 20 — — —	20 50 10 -1600 -40	$\mu\text{A}$

\* $V_{pp}$  is pin 6 on the MC68705P3 and is connected to  $V_{CC}$  in the normal operating mode. In the MC6805P2, pin 6 is NUM and connected to  $V_{SS}$  in the normal operating mode. The user must allow for this difference when emulating the MC6805P2 ROM based MCU.

\*\*Due to internal biasing, this input (when not used) floats to approximately 2.0 V.

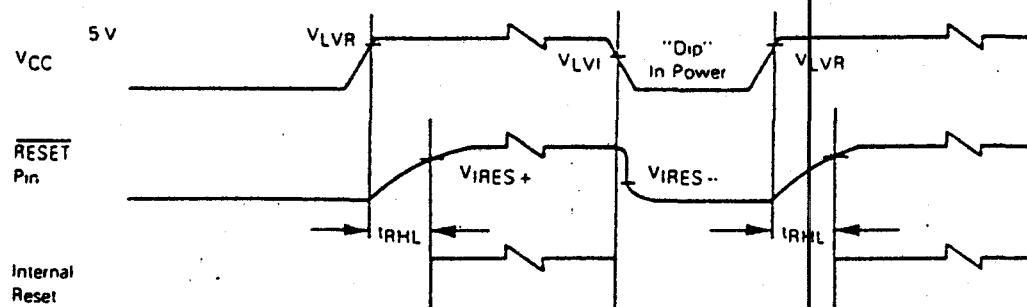


Figure 13. Power and Reset Timing

PORT DC ELECTRICAL CHARACTERISTICS ( $V_{CC} = +5.25 \pm 0.5$  Vdc,  $V_{SS} = 0$  Vdc,  $T_A = 0^\circ$  to  $70^\circ$ C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
<b>Port A</b>					
Output Low Voltage, $I_{Load} = 1.6$ mA	$V_{OL}$	—	—	0.4	V
Output High Voltage, $I_{Load} = -100$ $\mu$ A	$V_{OH}$	2.4	—	—	V
Output High Voltage, $I_{Load} = -10$ $\mu$ A	$V_{OH}$	$V_{CC} - 10$	—	—	V
Input High Voltage, $I_{Load} = -300$ $\mu$ A (Max)	$V_{IH}$	2.0	—	$V_{CC} + 0.7$	V
Input Low Voltage, $I_{Load} = -500$ $\mu$ A (Max)	$V_{IL}$	$V_{SS}$	—	0.8	V
Hi-Z State Input Current ( $V_{in} = 2.0$ V to $V_{CC}$ )	$I_{IH}$	—	—	-300	$\mu$ A
Hi-Z State Input Current ( $V_{in} = 0.4$ V)	$I_{IL}$	—	—	-500	$\mu$ A
<b>Port B</b>					
Output Low Voltage, $I_{Load} = 3.2$ mA	$V_{OL}$	—	—	0.4	V
Output Low Voltage, $I_{Load} = 10$ mA (Sink)	$V_{OL}$	—	—	1.0	V
Output High Voltage, $I_{Load} = -200$ $\mu$ A	$V_{OH}$	2.4	—	—	V
Darlington Current Drive (Source), $V_O = 1.5$ V	$I_{OH}$	-1.0	—	-10	mA
Input High Voltage	$V_{IH}$	2.0	—	$V_{CC} + 0.7$	V
Input Low Voltage	$V_{IL}$	$V_{SS}$	—	0.8	V
Hi-Z State Input Current	$I_{TSI}$	—	2	20	$\mu$ A
<b>Port C</b>					
Output Low Voltage, $I_{Load} = 1.6$ mA	$V_{OL}$	—	—	0.4	V
Output High Voltage, $I_{Load} = -100$ $\mu$ A	$V_{OH}$	2.4	—	—	V
Input High Voltage	$V_{IH}$	2.0	—	$V_{CC} + 0.7$	V
Input Low Voltage	$V_{IL}$	$V_{SS}$	—	0.8	V
Hi-Z State Input Current	$I_{TSI}$	—	2	20	$\mu$ A

SWITCHING CHARACTERISTICS ( $V_{CC} = +5.25 \pm 0.5$  Vdc,  $V_{SS} = 0$  Vdc,  $T_A = 0^\circ$  to  $70^\circ$ C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Oscillator Frequency Normal	$f_{osc}$	0.4	—	4.2	MHz
Instruction Cycle Time ( $4/f_{osc}$ )	$t_{cyc}$	0.950	—	10	$\mu$ s
INT or Timer Pulse Width (See Interrupt Section)	$t_{WL}, t_{WH}$	$t_{cyc} + 250$	—	—	ns
RESET Pulse Width	$t_{RWL}$	$t_{cyc} + 250$	—	—	ns
RESET Delay Time (External Cap = 1.0 $\mu$ F)	$t_{RHL}$	100	—	—	ms
INT Zero Crossing Detection Input Frequency	$f_{INT}$	0.03	—	1.0	kHz
External Clock Duty Cycle (EXTAL)	—	40	50	60	%

PROGRAMMING OPERATION ELECTRICAL CHARACTERISTICS ( $V_{CC} = +5.25 \pm 0.5$  Vdc,  $V_{SS} = 0$  Vdc,  $T_A = 20^\circ$  to  $30^\circ$ C, unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
Programming Voltage ( $V_{pp}$ Pin)	$V_{pp}$	20.0	21.0	22.0	V
$V_{pp}$ Supply Current	$I_{pp}$	—	—	8	mA
$V_{pp} = 5.25$ V		—	—	30	
$V_{pp} = 21.0$ V		—	—		
Programming Oscillator Frequency	$f_{oscp}$	0.9	1.0	1.1	MHz
Bootstrap Programming Mode Voltage (TIMER Pin) $I_{in} = 100$ $\mu$ A Max	$V_{IHTP}$	9.0	12.0	15.0	V

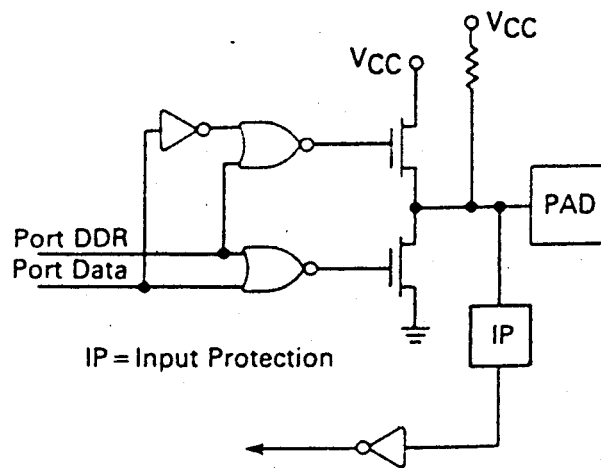


Figure 14. Port A Logic Diagram

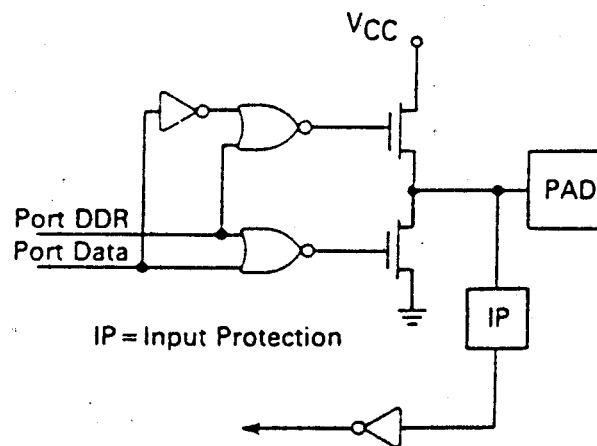


Figure 15. Port B and Port C Logic Diagram

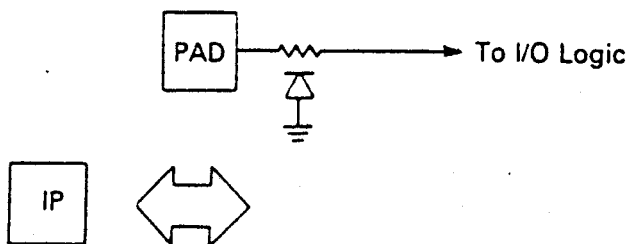


Figure 16. Typical Input Protection

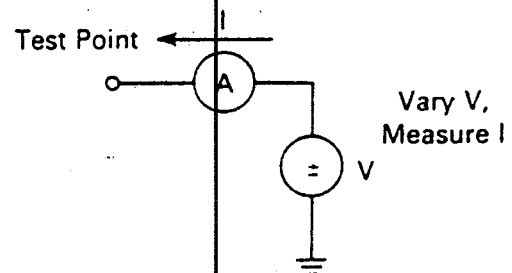


Figure 17. I/O Characteristic Measurement Circuit

## ORDERING INFORMATION

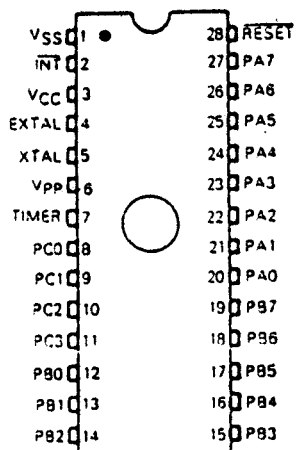
The following table provides generic information pertaining to the package type, temperature, and MC order numbers for the MC68705P3.

Table 3. Generic Information

Package Type	Internal Clock Frequency (MHz)	Temperature	Order Number
Cerdip (S Suffix)	1.0	0° to 70°C	MC68705P3S
Cerdip (S Suffix)	1.0	-40° to +85°C	MC68705P3CS

## MECHANICAL DATA

## PIN ASSIGNMENTS



## USULAN TUGAS AKHIR

- a. Judul Tugas Akhir : PERENCANAAN DAN PEMBUATAN PENGHITUNG KESALAHAN KWH METER DENGAN MENGGUNAKAN MIKROKONTROLER MC68705
- b. Ruang lingkup : - Pengukuran Listrik  
- Mikroprosesor  
- Bahasa Assembly
- c. Latar belakang : Perkembangan dan ilmu pengetahuan dan teknologi telah banyak menawarkan kemudahan-kemudahan bagi manusia dalam menjalankan aktifitasnya, baik dalam pekerjaan ringan maupun pekerjaan yang kompleks. Pemakaian mikroprosesor juga telah berkembang pesat dalam berbagai bidang, terutama dalam dunia industri. Banyak sekali kegunaan mikroprosesor ini, sehingga perkembangannya sangat pesat. Dalam penyaluran energi listrik untuk setiap pemakaian perlu dicatat atau direkam, sehingga akan diketahui berapa pemakaian energi listrik dalam jangka waktu tertentu. Mengingat akan pentingnya akan hal ini, maka diperlukan kWh meter yang ketelitiannya memenuhi standar yang telah ditentukan, karena akan menyangkut berapa besarnya biaya yang harus dibayar oleh konsumen. Pengecekan ketelitian kWh meter agar sesuai dengan standar yang telah ditentukan oleh PLN, biasanya dilakukan secara manual. Untuk memudahkan pembacaan kesalahan kWh meter tersebut, maka akan lebih mudah apabila kesalahannya dapat terbaca secara langsung (*displayed*). Dalam tugas akhir ini akan dibahas mengenai perencanaan dan pembuatan penghitung kesalahan kWh meter berbasis IC mikrokontroler MC 68705, dan didukung dengan kWh meter standar. Dengan alat ini diharapkan dapat mengetahui kesalahan kWh meter, baik untuk peneraan maupun pengontrolan di lapangan, karena alat ini mudah dibawa kemana-mana dan cara pemakaiannya mudah.
- d. Penelaahan studi : - Mempelajari prinsip kerja kWh meter dan standar peneraannya.  
- Mempelajari sifat-sifat dari setiap komponen yang digunakan.  
- Mempelajari sistem input output komponen.  
- Mempelajari bahasa program yang digunakan.

- e. Tujuan : Merencanakan dan membuat alat penghitung kesalahan kWh meter dengan menggunakan IC mikrokontroler MC 68705, sebagai komponen utama.
- f. Metodologi : 1. Studi literatur.  
2. Perencanaan rangkaian.  
3. Penyediaan komponen dan perakitan.  
4. Pembuatan program.  
5. Pengujian alat.  
6. Penyusunan laporan dan kesimpulan sebagai penyelesaian akhir.
- g. Relevansi : Dengan alat ini diharapkan dapat membantu perusahaan Umum Listrik Negara dan pihak yang memerlukan, dengan demikian kesalahan kWh meter dapat segera diketahui, baik untuk peneraan maupun pengontrolan di lapangan, karena alat ini mudah dibawa kemana-mana dan cara pemakaiannya mudah.
- h. Jadwal kegiatan :

KEGIATAN	BULAN					
	I	II	III	IV	V	VI
Persiapan						
Studi literatur						
Perencanaan rangkaian						
Penyediaan komponen						
Pembuatan program						
Pengujian alat						
Pembahasan						





DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN  
**JURUSAN TEKNIK ELEKTRO**  
**FAKULTAS TEKNOLOGI INDUSTRI**

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

KAMPUS : ITS - KEPUTIH SUKOLILO - TELP. (031) 5947302 FAX. 031 - 5947302 SURABAYA

Nomor : 104.A/PT.12.FTI.3/M/1994  
Lampiran : 1 ( Satu ) set.  
Hal : Fasilitas menggunakan  
Peralatan Laboratorium.

24 JUNI 1994

Kepada : Yth. Pemimpin PLN  
DISTRIBUSI JAWA TIMUR  
u.p. Kepala Siralab.  
Jl. Embong Wungu  
SURABAYA

Dengan ini kami hadapkan mahasiswa :

Nama : Kusno Utomo  
Nomor pokok : 2912201806  
Tahap pendidikan : Sarjana  
Bidang Studi : Teknik Sistem Tenaga

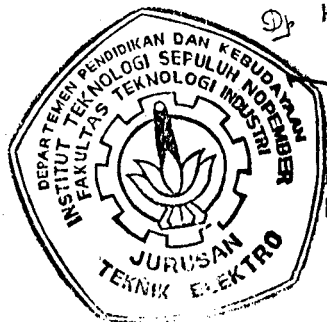
Sehubungan dengan Tugas Akhir yang dikerjakan oleh mahasiswa tersebut dengan judul :

PERENCANAAN DAN PEMBUATAN PENGHITUNG KESALAHAN KWH  
METER DENGAN MENGGUNAKAN MIKROKONTROLER MC 68705

maka kami mohon bantuan dan fasilitas menggunakan peralatan Laboratorium yang sangat di butuhkan untuk uji coba alat tugas akhir mahasiswa tersebut di atas . Apabila memungkinkan kami harapkan hal tersebut dapat digunakan mulai tanggal 01 JULI 1994 selama 1 (Satu) bulan.

Atas perhatian ,bantuan dan kerja sama yang berikan kami ucapkan terima kasih.

Ketua ,



Dr.Ir. Moch. Salehudin, M.Eng.Sc.  
NIP. 130 687 438.

Tembusan kepada :

Yth. Dekan FTI - ITS.

BOP/SIRALAB

Surabaya, 26 Juli 1994.

Kepada Yth. :  
Ketua Jurusan Teknik Elektro  
Fakultas Teknologi Industri ITS  
Jl. Arif Rahman Hakim  
S U R A B A Y A.-

Sesuai dengan surat Saudara No. : 104.A/PT.12.FTI.3/M/1994 tanggal 24  
Juli 1994 hal : Fasilitas menggunakan Peralatan Laboratorium, maka  
mahasiswa :

Nama : KUSNO UTOMO.  
Nomor pokok : 2912201806.  
Bidang Studi : Teknik Sistem Tenaga.

pada :

Hari : Sabtu.  
Tanggal : 16 Juli 1994.  
Di : BOP/Siralab  
PLN Distribusi Jawa Timur

telah melakukan pengujian alat :

PENGHITUNG KESALAHAN KWH METER DENGAN MENGGUNAKAN MIKROKONTROLER  
MC 68705

yang disaksikan oleh :

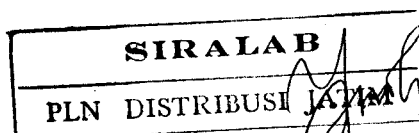
1. Bapak DR. Ir. SUSILO MATAIR dari ITS.
2. Sdr. SARNOTO dari BOP/Siralab PLN Distribusi Jawa Timur.
3. Sdr. ANWAR dari BOP/Siralab PLN Distribusi Jawa Timur.
4. Sdri. HERIYATI dari BOP/Siralab PLN Distribusi Jawa Timur.

dengan hasil pengujian :

Alat tersebut berfungsi dengan baik dan dengan ketelitian yang tinggi.

Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana  
mestinya.

BOP/SIRALAB  
PLN DISTRIBUSI JAWA TIMUR



Ir. WAHYUDIATI  
K. Siralab